



IDEC CORPORATION

IANF2 and IANF2E
2-axis Motion Controller
User's Manual

TABLE OF CONTENTS

About This Manual

Audience	7
Applicable Units	7
Navigating this Manual	7
Manual Conventions	7
Trademark Notices	7
Revision Record	7
Revision History	7
Manual Layout	8

Reference: IANF2(E) Specifications

Introduction	9
AMCI by IDEC Corporation	
Motion Control Products	9
AnyNET-I/O	9
Module Overview	9
Modes of Operation	10
Configuration Mode	10
Command Mode	10
Power Up Behavior	10
Specifications	11
Front Panel	12
Address Settings	12
Front Panel Status LED's	12
Power Connector	13
Network Port	13
Network Status LED's	13
I/O Connector	14
Digital Outputs	14
Set_State Bit	15
Digital Inputs	15
Home Input	15
CW Limit Switch or	
CCW Limit Switch	15
Emergency Stop Input	15
Capture Input	15
External Input	15
Encoder Inputs	15
Pulse Train Outputs	15

Reference: Move Profiles

Definitions	17
Units of Measure	17
Motor Position	17
Home Position	17
Valid and Invalid Positions	17
Count Direction	17
Starting Speed	18
Target Position	18
Relative Coordinates	18
Absolute Coordinates	18
Acceleration Types	18
Constant Acceleration	18

Reference: Move Profiles (continued)

Triangular S-Curve Acceleration	19
Trapezoidal S-Curve	
Acceleration	19
A Simple Move	19
Profile Equations	20
Controlled and Immediate Stops	20
Host Control	20
Hardware Control	20
Basic Move Types	21
Relative Move	21
Absolute Move	21
Jog CW/CCW Moves	22
Controlled Stops	23
Immediate Stops	23
Registration Move CW/CCW	23
Controlled Stops	24
Immediate Stops	24
Controlling Moves In Progress	25
Find Home Moves	25
Jog Moves	25
Registration Moves	25
Absolute and Relative Moves	25
Encoder Follower Move	25
Multiplier and Divisor	26
Acceleration/Deceleration	26
How It Works	26
Controlled Stop Conditions	26
Immediate Stop Conditions	26
Advanced Ratio Control	26

Reference: Calculating Move Profiles

Units of Measure	27
Linear Acceleration Equations	27
Variable Definitions	27
Total Time Equations	29
S-Curve Acceleration Equations	29
Triangular S-Curve Acceleration	29
Trapezoidal S-Curve	
Acceleration	30

Reference: Interpolated Move Profiles

The X-Y Plane	31
Linear Interpolated Moves	31
Relative Linear Move	31
Absolute Linear Move	31
Required Parameters	32
Circular Interpolated Moves	32
Center Point Method	32
Radius Method	33
Via Point Method	34
Controlled and Immediate Stops	34
Handling Rounding Errors	35

Reference: Homing the IANF2(E)

Definition of Home Position 37
 Preset Position 37
 Home CW/CCW Commands 37
 Homing Inputs 37
 Physical Inputs 37
 Backplane Inputs 37
 Homing Configurations 38
 Homing Profiles 38
 Home Input Only Profile 38
 Profile with Proximity Input 39
 Profile with Overtravel Limit 40
 Controlling Home CW/CCW
 Commands In Progress 40
 Controlled Stop Conditions 40
 Immediate Stop Conditions 40

Task: Installing the IANF2(E)

Safe Handling Guidelines 41
 Prevent Electrostatic Damage 41
 Prevent Debris From
 Entering the Unit 41
 Remove Power Before Servicing ... 41
 Mounting 41
 Dimensions 41
 Minimum Spacing 42
 DIN Rail Installation 42
 Installing IC-5
 Connectors (as needed) 42
 Mounting the IANF2(E) Module 42
 Addressing 43
 Ethernet Connections 43
 Power Wiring 44
 Required Power 44
 Wiring 44
 I/O Connector Pinout 45
 Drive Wiring 45
 Differential Wiring 45
 Open Collector Wiring
 (Sourcing Inputs on Drive) 46
 Open Collector Wiring
 (Sinking Inputs on Drive) 46
 Digital Output Wiring 47
 Discrete Input Wiring 47
 Sinking Output Sensors Require
 a Pull Up Resistor 48
 Encoder Wiring 48
 Differential Encoder Wiring 48
 Single Ended Wiring 49
 Diagnostic Feedback Wiring 50

Task: Set the IP Address

Determine the Best Method for
 Setting the IP Address 51
 Use Factory Default Settings 51
 Use the AMCI by IDEC Ethernet Tool 51

Task: WindLDR 8.5 Project Setup

Add the IANF2E to the
 Remote Host List 53
 Configure the Connection Settings
 for the IANF2E Stack 54

Task: Installing Custom Macros

Download the Custom Macro File 57
 Add the Macros to a New or
 Existing Project 57

Task: Using a Custom Macro in WindLDR

Verify Memory Usage 59
 Axis Addressing in an
 AnyNET-I/O Stack 59
 Verify/Alter Internal Relay
 Keep Settings 60
 Define the Memory Map to the
 Macro's Argument Devices 60
 Configure IANF2 61
 Absolute Move IANF2E 62
 Relative Move IANF2E 62
 Hold IANF2E 62
 Resume IANF2E 63
 Immediate Stop IANF2E 63
 Home CW IANF2E 63
 Home CCW IANF2E 63
 Jog CW IANF2E 64
 Registration Move CW IANF2E 64
 Jog CCW IANF2E 65
 Registration Move CCW IANF2E ... 65
 Preset Position IANF2E 65
 Clear Errors IANF2E 66
 Preset Encoder Position IANF2E ... 66
 Encoder Follower CW IANF2E 66
 Encoder Follower CCW IANF2E 66
 Set Min. Reg. Move Distance
 IANF2E 67
 Linear Interpolated Move
 IANF2E 67
 Via Point Circular Move IANF2E ... 68
 Center Point Circular Move
 IANF2E 69
 Radius Circular Move IANF2E 70
 Add a User-defined Macro to
 Ladder Logic 71

Task: Additional Logic and Input Data Formats

Axis Independence	73
Power-Up Behavior	73
Buffer Input Data	73
Command Bits Must Transition	73
Commands that do not cause motion	73
Jog and Registration Moves	73
Interpolated, Absolute, Relative, and Find Home Moves	73
Command Mode Input Data Format	74
Status Word 0	75
Status Word 1	76
Network Connectivity (Word 9)	77
Configuration Mode Input Data Format ...	78
Configuration Word 0 Mirror	79
Configuration Word 1 Mirror	80

Reference: Tabulated Command and Error Codes

Status Bits	81
Axis Stopped	81
Move Complete Bit	81
Home Invalid Bit	81
Profile Invalid Bit	81
Position Invalid Bit	82
Input Error Bit	82
Command Error Bit	82
Configuration Error Bit	83
Command Error Conditions	
By Command	83
Jog CW/CCW Command	83
Blend CW/CCW Command	83
Registration Move Command	83
Encoder Follower Operations	83

Reference: Config Mode Output Data Format

Power Up Behavior	85
Data Format	85
Output Data Format	86
Configuration Word 0	87
Configuration Word 1	88
Starting Speed	88
Homing Timeout	88
Invalid Configurations	88
Configuration Word 0	89
Configuration Word 1	89
Starting Speed Parameter	89
Homing Timeout Parameter	89
Reserved Words	89

Reference: Command Mode Output Data Format

Data Format	91
Command Bits Must Transition	91
Single Axis Commands	91
Command Word 0	92
Command Word 1	93
Single Axis Command Blocks	93
Absolute Move	93
Relative Move	94
Hold Move	94
Resume Move	94
Immediate Stop	95
Home CW	95
Home CCW	96
Jog CW	96
Registration Move CW	96
Encoder Follower CW	97
Jog CCW	97
Registration Move CCW	97
Encoder Follower CCW	98
Preset Position	98
Preset Encoder Position	99
Reset Errors	99
Blend Move CW	99
Blend Move CCW	100
Set Minimum Registration Move Distance	100
Interpolated Move Command Blocks	101
Linear Interpolated Move	101
Via Point Circular Interpolated Move	102
Center Point Circular Interpolated Move	103
Radius Circular Interpolated Move	104

Reference: Blend Moves

Blend Moves	105
Controlled Stop Conditions	106
Immediate Stop Conditions	106
Controlling a Blend Move In Progress	106
Programming Blend Move Profiles	106
Initial Blend Profile Programming Block	107
Secondary Profile Programming Block	108
Commanding a Blend Move	108

***Optional Task: Configure Your
Network Interfaces***

Firewall Settings	109
Disable All Unused Network Interfaces	109
Configure Your Network Interface	109
Test Your Network Interface	110

ABOUT THIS MANUAL

Introduction

Read this chapter to learn how to navigate through this manual and familiarize yourself with the conventions used in it. The last section of this chapter highlights the manual's remaining chapters and their target audience.

Audience

This manual explains the installation and operation of the IANF2 and IANF2E Two Axis Servo/Stepper Controller Module. It is written for the engineer responsible for incorporating the IANF2 or IANF2E into a design, as well as the engineer or technician responsible for its actual installation. If there are any unanswered questions after reading this manual, call the factory. An applications engineer will be available to assist you.

Applicable Units


This manual applies to IANF2 and IANF2E modules. The IANF2E contains an integral network connection that allows the IANF2E to connect itself, and up to five other modules, to an IDEC MicroSmart FC6A controller over a Modbus TCP network. When information in this manual applies equally to both units the term "IANF2(E)" is used.


Navigating this Manual


This manual is designed to be used in both printed and on-line forms. Its on-line form is a PDF document, which requires Adobe Acrobat Reader version 7.0+ to open it. You are allowed to select and copy sections for use in other documents and add notes and annotations. If you decide to print out this manual, all sections contain an even number of pages which allows you to easily print out a single chapter on a duplex (two-sided) printer.

Manual Conventions

Three icons are used to highlight important information in the manual:

 **Note** Notes highlight important concepts, decisions you must make, or the implications of those decisions.

 **Caution** Cautions tell you when equipment may be damaged if the procedure is not followed properly.

 **Warning** Warnings tell you when people may be hurt or equipment may be damaged if the procedure is not followed properly.

The following table shows the text formatting conventions:

Format	Description
Normal Font	Font used throughout this manual.
<i>Emphasis Font</i>	Font used for parameter names and the first time a new term is introduced.
<i>Cross Reference</i>	When viewing the PDF version of the manual, clicking on a blue cross reference jumps you to referenced section of the manual.
<i>HTML Reference</i>	When viewing the PDF version of the manual, clicking on a red cross reference opens your default web browser to the referenced section of the IDEC website if you have Internet access.

Trademark Notices

"FC6A Series MicroSmart" is a trademark of IDEC Corporation. The AMCI logo and "AnyNET-I/O" are trademarks of Advanced Micro Controls Inc. "Adobe" and "Acrobat" are registered trademarks of Adobe Systems Incorporated.

All other trademarks contained herein are the property of their respective holders.

Revision Record

This manual, 940-01100, is the first revision of this manual.

Revision History

940-01100: Initial Release.

Manual Layout

You will most likely read this manual for one of two reasons:

- If you are curious about the IANF2 or IANF2E products, this manual contains the information you need to determine if these products are the right products for your application. The first chapter, *IANF2(E) Specifications* contains all of the information you will need to fully specify the right product for your application.
- If you need to install and use an IANF2 or IANF2E product, then the rest of the manual is written for you. To simplify installation and configuration, the rest of the manual is broken down into *references* and *tasks*. Using an IANF2 or IANF2E product requires you to complete multiple tasks, and the manual is broken down into sections that give the background information you need to understand how to use IANF2(E) (references), and instructions to complete operations (tasks).

Section Title	Starting Page #	Section Description
<i>IANF2(E) Specifications</i>	9	Complete specifications for the IANF2 and IANF2E products.
<i>Move Profiles</i>	17	Reference information on how the IANF2(E) can be used to control single axis motion in your application.
<i>Interpolated Move Profiles</i>	31	Reference information on how the IANF2(E) can be used to control motion in your application that involves both axes.
<i>Homing the IANF2(E)</i>	37	Reference information on how to set the home position of the IANF2(E).
<i>Installing the IANF2(E)</i>	41	Task instructions covering how to install an IANF2(E) on a machine. Includes information on mounting, grounding, and wiring specific to the units.
<i>Set the IP Address</i>	51	Task instructions that covers the options for setting the IP address on an IANF2(E).
<i>WindLDR 8.5 Project Setup</i>	53	Task instructions for adding an IANF2(E) to a WindLDR 8.4+ project.
<i>Installing Custom Macros</i>	57	Task instructions for downloading and installing custom macros into a new or existing WindLDR 8.4+ project.
<i>Using a Custom Macro in WindLDR</i>	59	Task instructions for adding a custom macro to a ladder logic program.
<i>Additional Logic and Input Data Formats</i>	73	Task guidelines for adding the applications specific logic needed to control an IANF2(E) along with reference information on the format of the input data from the IANF2(E). Data formats for configuration and command responses are given.
<i>Calculating Move Profiles</i>	27	Reference information on calculating detailed move profiles.
<i>Config Mode Output Data Format</i>	85	Reference information on the format of configuration data written down to the IANF2(E). Custom macros simplify the programming of the IANF2(E), and should be used in most cases. This reference is provided as a troubleshooting aid should the need arise.
<i>Command Mode Output Data Format</i>	91	Reference information on the format of command data written down to the IANF2(E). Custom macros simplify the programming of the IANF2(E), and should be used in most cases. This reference is provided as a troubleshooting aid should the need arise.
<i>Blend Moves</i>	105	Instructions for the task of configuring the IANF2(E) to perform Blend Moves. At this point in time, User-defined Macros for this functionality are not available from IDEC.
<i>Configure Your Network Interfaces</i>	109	Reference information on error conditions and their causes.
<i>Configure Your Network Interfaces</i>	109	Instructions for the optional task of configuring network interfaces on your computer or laptop.

REFERENCE 1: IANF2(E) SPECIFICATIONS

Introduction

This chapter contains all of the information needed to understand how the FC6A MicroSmart controller interfaces with AMCI by IDEC Corporation products and specify IANF2 or IANF2E devices for your project.

AMCI by IDEC Corporation Motion Control Products

AMCI by IDEC Corporation is a partnership between the two companies to bring AMCI motion control products to the FC6A MicroSmart product family. Products that have been designed and manufactured by AMCI are factory configured to work immediately with the FC6A controllers. Setting the IP address is the only step needed before incorporating these products into your system. The IP address can be set with a freely available utility from the IDEC website.

In addition to the simplified hardware configuration, IDEC Corporation has released a collection of macros that simplify programming the FC6A MicroSmart controllers. In addition to these macros, IDEC has defined a set of Data Registers that are used by all sample programs for the motion control products. These Data Registers predefine twelve motion axes. With the exception of the IANF2(E), each AMCI by IDEC Corporation product controls one axis. This is true for the IANG1(E) as well as the stepper motors with integrated controllers. The IANF2(E) products control two axes. These modules use two consecutive sets of Data Registers to control their two axes.

AnyNET-I/O

The IANF2 and IANF2E mount on a DIN rail and can be connected together through an edge card connector that snaps into the DIN rail. Up to six modules can be connected together in an *AnyNET-I/O Stack*. The IANF2E contains an Ethernet port that is used to connect the stack to the Modbus TCP network. The IANF2 does not contain an Ethernet port, and can only function as a member of an *AnyNET-I/O Stack*.

Module types can be mixed within an AnyNET-I/O Stack. For example, a stack can contain an IANG1E module, two IANG1 modules and two IANF2 modules for a total of five modules controlling a total of seven motion axes. When the FC6A is configured, the IP address assigned to the IANG1E would be given enough data registers to communicate with the seven axes.

Module Overview

The IANF2(E) modules offer one or two independent motion control axes that function in an open-loop configuration. Each axis can be used to drive a stepper or a servo with step/direction input capability. The IANF2(E) can also synchronize its two axes, giving you the ability to control linear and circular motions in an XY plane.

The IANF2(E) requires twenty input data registers and twenty output data registers in the FC6A host controller.

Each axis of the IANF2(E) offers a full 32 bit (± 2 billion+) motor position register, move lengths of up to 2^{31} (± 1 billion+) counts, programmable S-curve acceleration types, five discrete inputs for various control functions and a differential encoder input.

The IANF2(E) has the following I/O connections for each axis:

- Differential outputs to your servo or stepper driver. Maximum output frequency is 1 MHz.
- Single ended outputs to your servo or stepper driver. Use these outputs if your driver does not have differential inputs. Maximum output frequency is 100 kHz. You cannot use the differential and single-ended outputs at the same time.
- Differential quadrature encoder inputs. An encoder can be used by the IANF2(E) for open loop position feedback or in *Encoder Follower Mode*. In Encoder Follower Mode, the IANF2(E) will generate pulses on the outputs in response to changes on the encoder inputs. This allows you to use the IANF2(E) in electronic gearing applications. The IANF2(E) also allows you to use the Z input on the encoder to define the *Home Position* of the machine. The Home Position is a reference position on the machine that you must define before running one type of move available with the IANF2(E).
- Home Input. This single ended DC input is typically used when defining the home position on the machine.
- CW Limit and CCW Limit Inputs. These single ended inputs are used to define the maximum clockwise and counter-clockwise positions on the machine. If one of these inputs becomes active while traveling in that direction, the IANF2(E) will immediately stop the move.



Figure R1.1 IANF2E Module

- Emergency Stop Inputs. If one or both of the CW and CCW Limits are not required, then the input can be configured as an Emergency Stop input. The IANF2(E) will immediately stop the move if an Emergency Stop input becomes active.
- Capture Input. This single ended input can be used to capture the motor position or encoder position during a move. This is useful in applications where you must capture the position value for an event that is too short to be captured by the FC6A controller.
- External Input. A single ended input that can be used to bring moves to a controlled stop.
- Three Digital Outputs per axis. These single ended outputs allow you externally monitor the status of the module. They can also be monitored and controlled by the FC6A host controller through the output data registers.

Modes of Operation

Each axis of the IANF2(E) has two modes of operation, *Configuration Mode* and *Command Mode*.

Configuration Mode

This mode give you that ability to configure the axis for your application without having to set any programming switches. It also allows you to change the axis configuration at any time. The axis must be configured after every power up. The following parameters are set in Configuration Mode:

- The enabled or disabled state of the axis. The second axis of an IANF2(E) is disabled by default.
- The enabled or disabled state of each of the five inputs
- The configuration of the CW Limit and CCW Limit as overtravel limits or Emergency Stop inputs
- The enabled or disabled state of the Backplane Home Proximity bit
- If a quadrature encoder will be used or not
- If Diagnostic Feedback will be used to test the IANF2(E) axis
- The Output Pulse type (CW/CCW or Step & Direction)
- If a homing operation will be to the Home Input or the encoder marker pulse
- The Starting Speed of most moves available with the IANF2(E). This parameter allows you to start a move above the motor's low frequency resonance point and is also useful in micro-stepping applications.
- The Homing Timeout value. This parameter allows you to automatically stop a homing operation if it takes longer to occur than expected.

Command Mode

This mode give you the ability to set the machine's home position, program, carry out, and control, servo or stepper motor moves, detect and reset errors as they occur, and run diagnostics on the I/O if needed.

- Allows you to set the home position in the machine
- Allows you to preset the motor position register
- Allows you to preset the encoder position register
- Allows you to run six different single-axis move types
 - Relative Move ➤ Absolute Move
 - Jog ➤ Registration Move
 - Blend Move ➤ Encoder Follower
 See reference section 2: *Move Profiles*, starting on page 17, for a complete description of these moves.
- Allows you to run Linear and Circular Interpolated Moves.
See Reference section 4: *Interpolated Move Profiles*, starting on page 31, for a complete description of these moves.
- Allows you to hold and resume relative and absolute moves
- Allows you to bring a move to an immediate stop via a backplane bit or hard-wired input
- Allows you to reset command errors so a new command can be issued
- Allows you to feed the driver outputs into the encoder inputs and perform diagnostics.

Power Up Behavior

The IANF2(E) will always power up in Command Mode and show a configuration error. Configuration data must be written to the IANF2(E) before commands can be issued to the module.

Specifications

Physical Dimensions

Width: 1.8 inches max.

Depth: 4.5 inches max.

Height: 3.9 inches

5.0 inches min. with mating connectors

Weight

0.43 lb. (0.19 kg) with mating connectors

Number of Axes

Two axes. Can be operated independently or synchronized for linear and circular interpolated moves.

Motor Control Outputs

Differential and single-ended outputs available.

Programmable CW/CCW or Step & Direction formats

Discrete Inputs

Five per Axis

Home, Capture, CW Limit, CCW Limit, and External

CCW Limit and CCW Limit inputs can be programmed to act as Emergency Stop inputs.

Discrete Outputs

Three per Axis

Output state controlled by IDEC controller. Outputs will also transition on move complete, input error, or on-line fault events.

Encoder Inputs

One Encoder per Axis

+5Vdc differential ABZ inputs

X4 decoding is always used by IANF2(E)

Maximum pulse train frequency: 250KHz
(1MHz maximum counting frequency)

Encoder can be used for position feedback, homing to Z pulse, and with Encoder Follower Mode.

Number of I/O Words per Module

IANF2(E): 20 Input registers, 20 Output registers

Differential Stepper Outputs

+5Vdc differential (3.5Vdc to 4Vdc peak typical)

Maximum Output Current: 20mA

Maximum Output Frequency: 1MHz

Single-Ended Stepper Outputs

Open Collector Sinking

Voltage Range: 12Vdc to 24Vdc

Maximum Output Current: 30mA

Maximum Output Frequency: 100KHz

An external supply is required for operation.

Digital Outputs

Same electrical specifications as the Single-Ended Stepper Outputs

Digital Inputs

Open Collector Sinking

All inputs share a common return

ON Voltage Range: 8Vdc to 24Vdc

OFF Voltage Range: 0Vdc to 5Vdc

Maximum Input Current: 15mA @ 24Vdc

An external supply is required for operation.

Throughput Time

100 microseconds

Inputs must remain in a steady state for 100 microseconds after a transition for the transition to be recognized.

Programming Storage

RAM Memory.

IANF2(E) must be configured on power up.

Power Supply

10 to 28 Vdc, surge to 30Vdc without damage to module. 120 mA @ 24Vdc (2.9W)

Environmental Conditions

Operating Temperature:-4 to 122°F
(-20 to 50°C)

Relative Humidity:5 to 95%
(w/o condensation)

Storage Temperature:-40 to 185°F
(-40 to 85°C)

Connectors

Mating connectors are supplied with the module and are also available separately under the following part numbers.

Connector	Part #	Wire	Strip Length	Min. Tightening Torque
I/O	IMS-2x11	28 - 16 AWG	0.275 inches	Spring Cage Connector
Power	IMS-4M	28 - 12 AWG	0.394 inches	4.43lb-in (0.5 Nm)
Backplane	IIC-5			

Front Panel

The front panels of two IANF2(E) modules are shown in figure R1.2. The clear front cover that covers the address switches and status LED's is hinged on the bottom. It swings down, and allows you to change the module's address in the AnyNET-I/O Stack with the address switches. The front panel also has the Status LED's, which give you information on the state of the module.

Address Settings

The AnyNET-I/O platform allows you to connect up to six modules to a single network connection in what is called an AnyNET-I/O Stack. The DIP switch behind the front panel cover is used to set the address of the module within the AnyNET-I/O Stack. The first module acts as the network interface and must have an address of zero. This address is set by having all of the DIP switches in their OFF position. (If you are using a single module, then it must have an address of zero.) The remaining modules in the Stack should have their addresses set to their position in the stack by setting the corresponding DIP switch to its ON position. Figure R1.2 shows the correct addressing for two modules. The module on the left is an IANF2E and has its address set to zero. The remaining module can be either an IANF2 or IANF2E module and its address is set to one

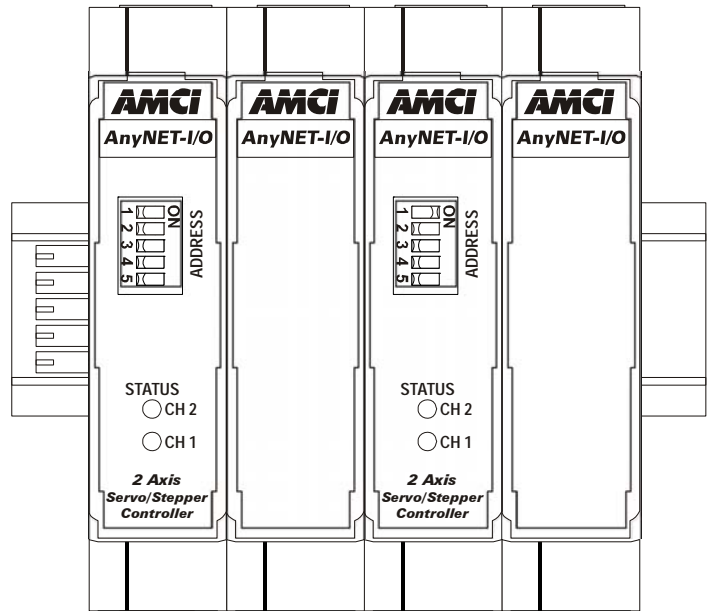


Figure R1.2 Status LED's

Note ▶ If an IANF2E has its address set to any value other than zero, its network interface is disabled. This allows you to use multiple IANF2E modules in a single AnyNET-I/O Stack.

Front Panel Status LED's

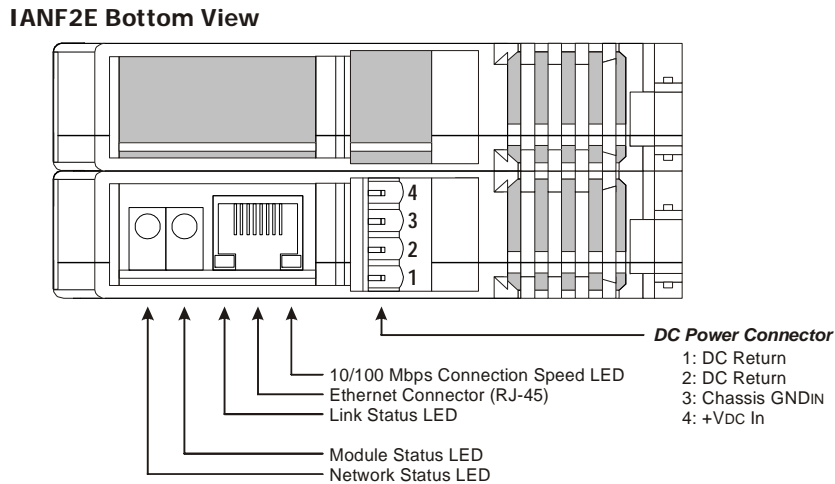
Each axis has a bi-color red/green status LED on the front of the module.

- ▶ **Steady Green:** Module OK, no motion on the axis is in progress.
- ▶ **Blinking Green:** Module OK, motion on the axis is in progress.
- ▶ **Blinking Red:** Configuration Error, Command Error, Input Error, Home Invalid Error, Invalid Profile Error.
- ▶ **Steady Red:** Communications failure.
 - ▶ AnyNET-I/O Stack has lost communication with the host controller.
 - ▶ Communications error between modules in the AnyNET-I/O Stack.
 - ▶ Communications error between the main processor and the communications co-processor on the AnyNET-I/O module that is acting as the network adapter

You must cycle power to the module to attempt to clear this fault.

Power Connector

The power connector of the IANF2(E) is located on the bottom of the module. Figure R1.3 below shows the location of the power connector and its pinout.



Network Port

Figure R1.3 also shows the location of the Ethernet port on an IANF2E module. This space is empty on an IANF2 module. The port is a standard RJ45 connector and supports 10/100 Mbps communications speeds. The port accepts shielded or unshielded CAT 5, 5e, or 6 cable. The port has an “auto switch” capability which eliminates the need for a crossover cable when directly connecting the module to a PC. A standard Ethernet cable can be used when connecting the IANF2(E) to any device.

Network Status LED's

The network status LED's indicate the health of the network connection between the IANF2E and its host.

- **Network Status** – Indicates the number of TCP connections to the IANF2E module. This LED will briefly flash red-green on power up while the module is initializing. It will flash green when a TCP connection is made to the driver. The number of flashes indicates the number of active connections. IANF2E modules support a maximum of three concurrent connections. There is a two second pause between flashes to make it easy to count the number of connections. If the LED is off, then there are no TCP connections to the module.
- **Module Status** – Indicates the health of the Ethernet co-processor on the module. This LED will briefly flash red once on power up and then flash green while the unit is initializing. It will be on steady green when the Ethernet co-processor is functioning correctly. It will turn red on an error. If this occurs, cycle power to the module.
- **Link Status** – On when there is a physical link between the Ethernet port of the AnyNET-I/O module and the Ethernet port of the device the module is plugged into. This LED will flash when data is being transmitted over the Ethernet link.
- **Connection Speed** – On when the Ethernet connection speed is 100Mbps. Off when the connection speed is 10 Mbps.

I/O Connector

Figure R1.4 below shows the pin out of the I/O connectors on the IANF2(E) modules. All mating connectors are included with the module. Additional connectors can be order for IDEC under the part number IMS-2x11.

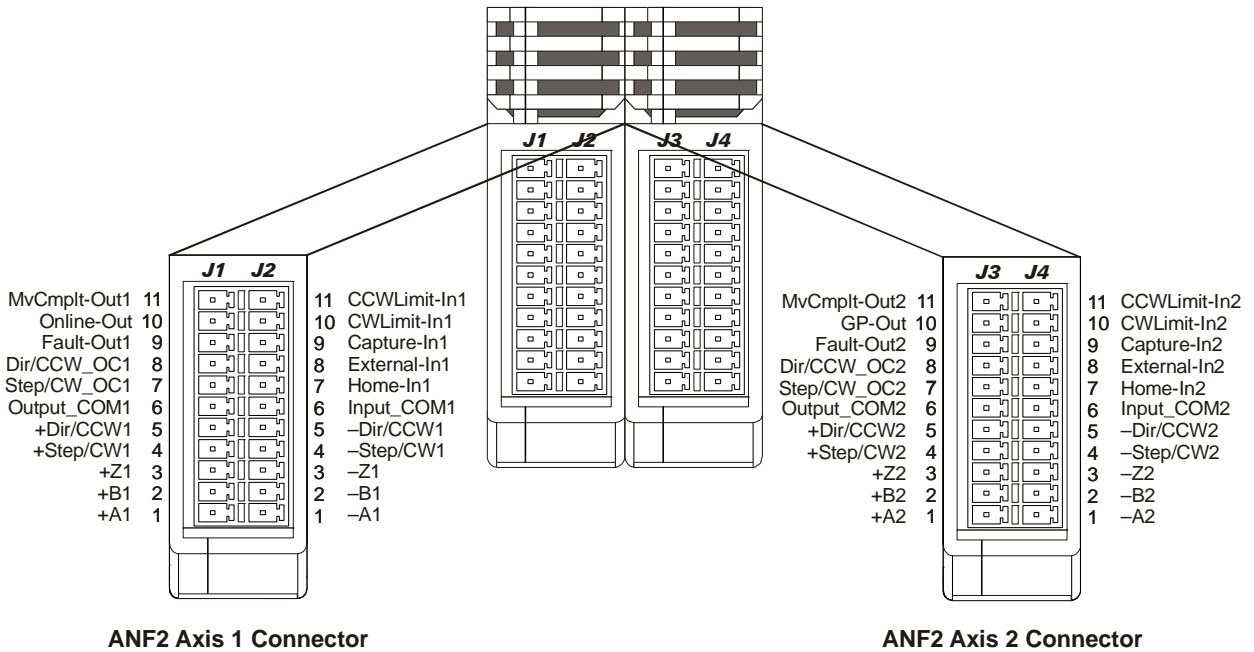


Figure R1.4 I/O Connector Pin Out

Digital Outputs

There are three DC digital outputs per axis on the IANF2(E) modules. The state of the outputs can be controlled and monitored by the IDEC controller. All outputs are open collector sinking. An output is connected to the Output_COM pin when the output is active. The Output_COM pin must be connected to the output power supply common for operation. The output power supply must be within the range of 5 to 24 Vdc.

An output is "active" when it is conducting current. The output state is reported to the IDEC controller as a "1" when the output is active. When measuring the output voltage on the pin, it will be near zero volts when the output is active.

No_Fault Output: (Axis 1 and 2). This output can only be set to its active state by the IDEC controller. Once active, this output will be turned off by the following conditions:

- Input Error during a move. (E-Stop, CW Limit, or CCW Limit Input activated during move.)
- Axis switched from Command to Configuration Mode
- Loss of network communications
- Reset command sent by host controller.

Move_Complete Output: (Axis 1 and 2) This output will be set to its active state by the following conditions:

- Absolute, Relative, Blend, or Interpolated Move completes without error.
- Set command sent by host controller.

Once active, this output can only be reset to its inactive state by the IDEC controller.

On_Line Output: (Axis 1 only) This output can only be set to its active state by the IDEC controller. The On_Line output will be turned off by the following conditions:

- Loss of network communications
- Reset command sent by host controller.

General Purpose (GP) Output: (Axis 2 only) The on/off state of the General Purpose output is only controlled by the IDEC controller.

Set_State Bit

It is possible, and suggested, to configure the IDEC FC6A controller to continuously write data to the IANF2(E). Because of this, the IANF2(E) implements a "Set_State" bit. Each axis has its own Set_State bit. Output states can only be changed by the IDEC controller when this bit equals "1".

- Note**►
- 1) All three outputs associated with the axis are updated when the associated Set_State bit equals "1". Your controller logic must take this into consideration when updating the state of the outputs.
 - 2) With the exception of the General Purpose output, the Set_State bit cannot be used to force the outputs on or off. When a No_Fault, Move_Complete, or On_Line event occurs, the outputs will change to their designated states for a maximum of five milliseconds before returning to the states specified by the IDEC controller.
 - 3) The Set_State bits should be reset during a move operation. This will prevent the IDEC controller from overriding the No_Fault and Move_Complete outputs while a move is occurring.

Digital Inputs

Each axis of the IANF2(E) has five discrete 12 to 24Vdc DC inputs. They are open collector sinking inputs. Their active states are fully programmable by the FC6A controller. Therefore, they can be used with Normally Open (NO) or Normally Closed (NC) outputs from the attached sensor.

Home Input

Many applications require that the machine be brought to a known position before normal operation can begin. This is commonly called "homing" the machine or bringing the machine to its "home" position. This input can be used with commands from the FC6A controller to set the home position. How the IANF2(E) uses the Home Input is described in reference chapter 5: *Homing the IANF2(E)*, starting on page 37.

CW Limit Switch or CCW Limit Switch

These inputs are used to define the limits of mechanical travel. For example, if you are moving in a clockwise direction and the CW Limit Switch activates, all motion will immediately stop. At this point, you will only be able to jog in the counter-clockwise direction. These inputs can also be defined as Emergency Stop Inputs if your application does not need one or both of them as limit inputs.

Emergency Stop Input

When an input is defined as an Emergency Stop, or E-Stop Input, the pulse train outputs will immediately stop when this input becomes active. A move cannot begin while this input is active. A separate Emergency Stop input does not exist on the module. The module must be configured to use either the CW or CCW Limit Switch input as an Emergency Stop Input.

Capture Input

This input is used to capture the current motor or encoder position value when it makes an inactive-to-active state transition. The motor position is captured if the encoder inputs were not enabled when the IANF2(E) are configured. The encoder position is captured if the encoder inputs were enabled when the IANF2(E) was configured.

External Input

An inactive-to-active transition on this input will bring some move types to a controlled stop. A controlled stop is one in which the programmed deceleration is used to bring the move to a stop. This is different from an immediate stop, where the output pulses are simply stopped, without deceleration.

Encoder Inputs

Each axis has three, 5 Vdc, differential inputs for a quadrature incremental encoder. Single ended sinking or sourcing encoders can also be used with the IANF2(E). If the encoder outputs exceed 5 Vdc, a current limiting resistor must be used. The A/B inputs have a maximum input frequency of 250 kHz. The IANF2(E) always uses 4X decoding, so the maximum count frequency is 1 MHz. The Z input can be used to home the machine as described in reference chapter 5: *Homing the IANF2(E)*, starting on page 37.

Pulse Train Outputs

Each axis has differential and open collector sinking outputs for connection to a servo or stepper motor drive. Differential is the preferred connection, with a maximum pulse train output frequency of 1 MHz. The open collector outputs should only be used when the servo or stepper motor drive does not support differential inputs. The maximum pulse train output frequency for the sinking outputs is 100 kHz. Each axis of the IANF2(E) can be configured to output pulses with the CW/CCW format or the STEP/DIR format.

Notes

REFERENCE 2: MOVE PROFILES

Introduction

When a move command is sent to an IANF2(E), the unit calculates the entire profile before starting the move or issuing an error message. This chapter explains how the single axis profiles are calculated and the different moves that are available.

Definitions

Units of Measure

Distance: Every distance is measured in steps. One output pulse from the IANF2(E) produces one step. When you configure your motor's drive, you will specify the number of steps you want to complete one rotation of the motor shaft. It is up to you to determine how many steps are required to travel the appropriate distance in your application.

Speed: All speeds are measured in steps/second. Since the number of steps needed to complete one shaft rotation is determined by your drive setup, it is up to you to determine how many steps per second is required to rotate the motor shaft at your desired speed.

Acceleration: The typical unit of measure for acceleration and deceleration is steps/second/second, or steps/second². However, when programming an IANF2(E), all acceleration and deceleration values must be programmed in the unit of measure of steps/millisecond/second.

- To convert from steps/second² to steps/millisecond/second, divide the value by 1000. This must be done when converting from a value used in the equations to a value programmed into an IANF2(E).
- To convert from steps/millisecond/second to steps/second², multiply the value by 1000. This must be done when converting from the value programmed into an IANF2(E) to the value used in the equations.

Motor Position

Motor Position is defined in counts, and its range is -2,147,483,648 to +2,147,483,647 counts. Clockwise moves will increase the motor position and counter-clockwise moves will decrease the motor position.

Home Position

The Home Position is any position on your machine that you can sense and stop at. There are two ways to defining the Home Position. The first is using the Preset Position command to set the Motor Position register to a known value. The second method is using one of the *Find Home* commands. If you use the unit's *Find Home* commands, the motor position and encoder position registers will automatically be set to zero once the home position is reached. Defining a Home Position is completely optional. Some applications, such as those that use the IANF2(E) for speed control, don't require position data at all.

Valid and Invalid Positions

The Motor Position is considered *Valid* once the Home Position has been defined. Until that time, the motor position is considered *Invalid*. Absolute moves cannot be run unless the position is Valid. Uncontrolled, immediate stops will force the position to become Invalid.

Count Direction

Move commands will order the IANF2(E) to move to a specific position, or move as long as the command is active. Moves to a position will output clockwise pulses when the motor position increases and output counter-clockwise pulses when the motor position decreases. Clockwise moves will increase the motor position and counter-clockwise moves will decrease the motor position.

		Clockwise Move (Increasing Count)	Counter-clockwise Move (Decreasing Count)
CW/CCW Outputs	CW		
	CCW		
Step & Direction Outputs	Step		
	Direction		

Starting Speed

The Starting Speed is the speed that most moves will begin and end at. In the IANF2(E), the Starting Speed parameter defines the output pulse frequency that all moves will begin and end at. This value is set while configuring the unit and it has a valid range of 1 to 1,000,000 steps/second. This value is typically used to start the move above the motor's low frequency resonances and, in micro-stepping applications, to limit the amount of time needed for acceleration and deceleration. A default value is not specified in this manual because it is dependent on motor driver, motor size, the number of steps per turn, and the attached load. While bench testing the IANF2(E), a starting speed between 0.25 and 0.5 RPS is generally a safe value to begin with.

Target Position

The Target Position is the position that you want the move to end at. There are two ways to define the Target Position, with relative coordinates or absolute coordinates.

Relative Coordinates

Relative coordinates define the Target Position as an offset from the present position of the motor. Most IANF2(E) moves use relative coordinates.

- The range of values for the Target Position when it is treated as an offset is $\pm 1,073,741,823$ counts. Positive offsets will result in clockwise moves, while negative offsets result in counter-clockwise moves.
- The Motor Position value reported back to the host can exceed $\pm 1,073,741,823$ counts. The only way to move beyond $\pm 1,073,741,823$ counts is with multiple relative moves or jog commands.

Absolute Coordinates

Absolute coordinates treat the Target Position as an actual position on the machine. Note that you must set the Home Position on the machine before you can run an Absolute Move. (See the definition of [Home Position](#) on the previous page.)

- The range of values for the Target Position when it is treated as an actual position on the machine is $\pm 1,073,741,823$ counts. The move will be clockwise if the Target Position is greater than the Current Position and counter-clockwise if the Target Position is less than the Current Position.
- The Motor Position value reported back to the host can exceed $\pm 1,073,741,823$ counts. However, you cannot move beyond $\pm 1,073,741,823$ counts with an Absolute Move. The only way to move beyond $\pm 1,073,741,823$ counts is with multiple relative moves or jog commands.

Acceleration Types

Most of the move commands allow you to define the acceleration type used during the move. The IANF2(E) supports three types of accelerations and decelerations. The type of acceleration used is programmed as part of the move. Note that the move's acceleration and deceleration will have the same type.

Constant Acceleration

With Constant Acceleration, the axis accelerates (or decelerates) at a constant rate until the programmed speed is reached. This offers the fastest acceleration, but consideration must be given to ensure the smoothest transition from rest to the acceleration phase of the move. The smoothest transition occurs when the configured Starting Speed is equal to the square root of the programmed Acceleration value. Note that other values will work correctly, but you may notice a quick change in velocity at the beginning or end of the acceleration phase.

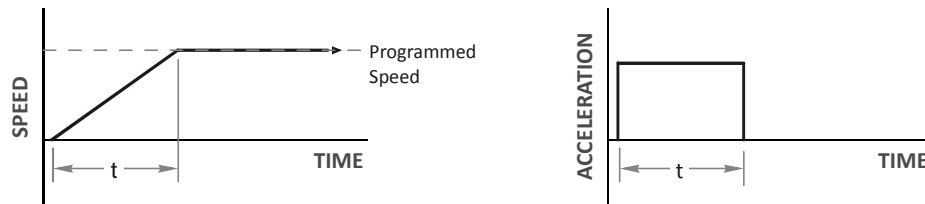


Figure R2.1 Constant Acceleration

Triangular S-Curve Acceleration

Triangular S-Curve Acceleration does not have a period of constant acceleration. The acceleration value increases to its programmed value during the first half of the acceleration phase, and decreases during the second half of the phase. Triangular S-Curve is the smoothest form of acceleration, but the time needed to reach the move's programmed speed is twice that of Constant Acceleration.

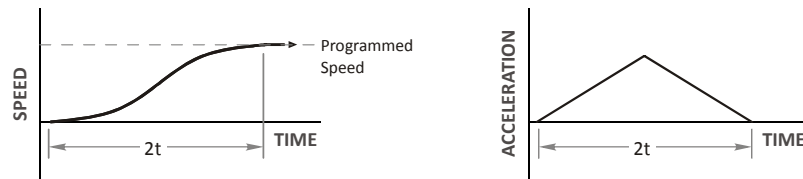


Figure R2.2 Triangular S-Curve Acceleration

Trapezoidal S-Curve Acceleration

Trapezoidal S-Curve acceleration has a period of constant acceleration in the middle of the acceleration phase. This constant acceleration section lasts for one half of the total acceleration time. The acceleration value increases until it reaches the value of the Acceleration Parameter. At this point, the acceleration remains constant until the IANF2(E) decreases the acceleration value until it equals zero when the programmed maximum speed is reached. Figure R2.3 shows a trapezoidal curve. With this setting, the Trapezoidal S-Curve acceleration only requires 33% more time to achieve the same velocity as a Constant Acceleration move.

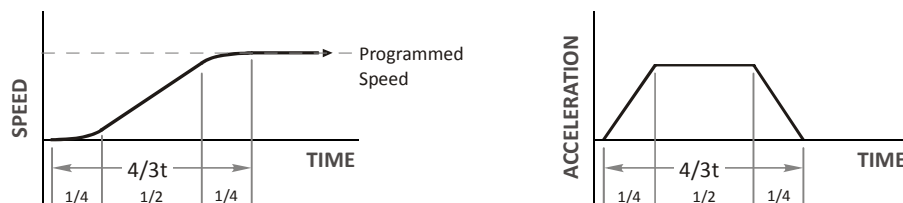


Figure R2.3 Trapezoidal S-Curve Acceleration

Additional information, including example move calculations, can be found in reference chapter 8, [Calculating Move Profiles](#) starting on page 27.

A Simple Move

As shown in the figure below, a move from A (Current Position) to B (Target Position) consists of several parts.

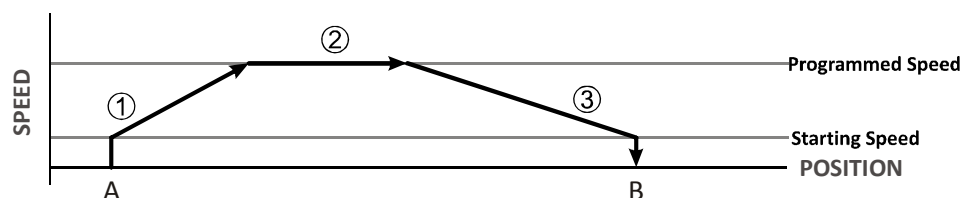


Figure R2.4 A Trapezoidal Profile

- 1) The move begins at point A, where the output frequency jumps from zero to the configured *Starting Speed*. The frequency of the pulse train then accelerates at the programmed *Acceleration Value* until the frequency of the pulse train reaches the *Programmed Speed*. Both the *Acceleration Value* and the *Programmed Speed* are programmed when the move command is sent to the IANF2(E).
- 2) The IANF2(E) continues to output pulses at the *Programmed Speed* until it reaches the point where it must decelerate before reaching point B.
- 3) The pulse train frequency decelerates at the *Deceleration Value*, which is also programmed by the move command, until the output frequency reaches the *Starting Speed*, which occurs at the *Target Position (B)*. The pulse train stops at this point. Note that the acceleration and deceleration values can be different in the move.

Figure R2.4 above shows a Trapezoidal Profile. A Trapezoidal Profile occurs when the *Programmed Speed* is reached during the move. This occurs when the number of pulses needed to accelerate and decelerate are less than the total number of pulses in the move.

Figure R2.5 below shows a Triangular Profile. A Triangular Profile occurs when the number of pulses needed to accelerate to the Programmed Speed and decelerate from the Programmed Speed are greater than the total number of pulses in the move. In this case, the profile will accelerate as far as it can before decelerating and the Programmed Speed is never reached.

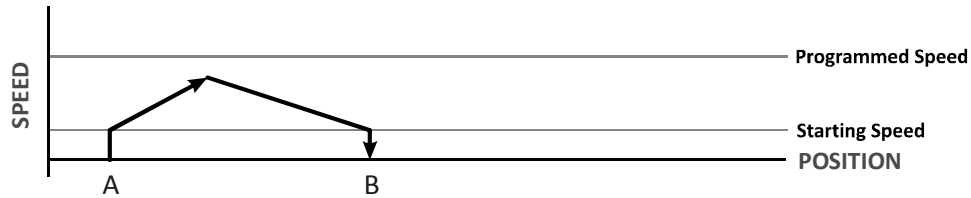


Figure R2.5 A Triangular Profile

Profile Equations

If your application requires very precise profiles, refer to reference chapter 8, [Calculating Move Profiles](#) for information on time and distance formulas.

Controlled and Immediate Stops

Once a move is started, there are several ways to stop the move before it comes to an end. These stops are broken down into two types:

- **Controlled Stop:** The axis immediately begins decelerating at the move's programmed deceleration value until it reaches the configured Starting Speed. The axis stops at this point. The motor position value is still considered valid after a Controlled Stop and the machine does not need to be homed again before Absolute Moves can be run.
- **Immediate Stop:** The axis immediately stops outputting pulses regardless of their output frequency. Under these conditions, it is possible for the inertia of the load attached to the motor to pull the motor beyond the stopping point. Therefore, the motor position value is considered invalid after an Immediate Stop and the machine must be homed again before Absolute Moves can be run.

Host Control

Hold Move Command: This command can be used with some moves to bring the axis to a Controlled Stop. The move can be resumed and finished, or it can be aborted. Not all moves are affected by this command. The section [Basic Move Types](#), starting below, describes each move type in detail, including if the move is affected by this command.

Immediate Stop Command: When this command is issued from the host, the axis will come to an Immediate Stop. The move cannot be restarted and the machine must be homed again before Absolute Moves can be run.

Hardware Control

External Input: The External Input can be used with some moves to bring the axis to a Controlled Stop. Not all moves are affected by this input. The section [Basic Move Types](#), starting below, describes each move type in detail, including if the move is affected by this input.

CW Limit and CCW Limit Inputs: In most cases, activating these inputs during a move will bring the axis to an Immediate Stop. The exceptions are the Find Home CW/CCW commands, the Jog Move CW/CCW commands, and the Registration Move CW/CCW commands. The Find Home CW/CCW commands are explained in reference chapter 5: [Homing the IANF2\(E\)](#), which starts on page 37. The commands for [Jog CW/CCW Moves](#) are fully explained on page 22, and the [Registration Move CW/CCW](#) commands are fully explained on page 23.

Emergency Stop Input: It is possible to configure the CW Limit Input and/or the CCW Limit Input as an Emergency Stop Input. When an Emergency Stop Input is activated, the axis will come to an Immediate Stop, regardless of the direction of travel.

Basic Move Types

Relative Move

Relative Moves move an offset number of pulses (n) from the Current Position (A). A trapezoidal profile is shown to the right, but Relative Moves can also generate triangular profiles. The command's Target Position is the move's offset. The offset can be in the range of $\pm 1,073,741,823$ counts. Positive offsets will result in clockwise moves, while negative offsets result in counter-clockwise moves.

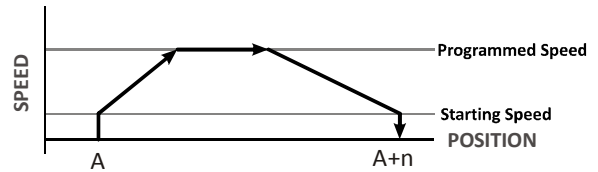


Figure R2.6 Relative Move

- Note** ➤
- 1) You do not have to preset the position or home the machine before you can use Relative Moves. That is, the state of the Position_Invalid status bit can be ignored.
 - 2) Relative Moves allow you to move your machine without having to calculate absolute positions. If you are indexing a rotary table, you can perform a relative move of 30° multiple times without recalculating new target positions in your controller. If you perform the same action with Absolute Moves, you would have to calculate your 30° position followed by your 60° position, followed by your 90° position, etc.

Controlled Stop Conditions

- The Jog Move Command bit is reset to "0".
- The Hold Move control bit in the command data from the FC6A controller makes a 0→1 transition. This transition will bring the axis into its Hold State. The use of the Hold Move bit, the Resume Move bit, and the External Input is further explained in the [Controlling Moves In Progress](#) section starting on page 25.
- The External Input for the axis makes an inactive-to-active transition. This transition will bring the axis into its Hold State, with the axis acting as if the Hold Move control bit in the command data from the FC6A controller made a 0→1 transition. The use of the External Input, the Hold Move bit, and the Resume Move bit is further explained in the [Controlling Moves In Progress](#) section starting on page 25.

If the External Input is active when a Relative Move is started, the IANF2(E) will output one step in the specified direction before entering the Hold State.

Immediate Stop Conditions

- The Immediate Stop bit makes a 0→1 transition in the command data from the FC6A controller.
- An inactive-to-active transition on an input configured as an E-Stop Input.
- A CW/CWW Limit Switch is reached.

Note ➤ An Input_Error bit is latched in the Network Input Data when an E-Stop or Limit Switch is activated. The next move command will clear this bit automatically if none of the inputs are active when the command is issued.

Absolute Move

Absolute Moves move from the Current Position (A) to a given position (B). (The IANF2(E) calculates the number of pulses needed to move to the given position and outputs that number of pulses.) A trapezoidal profile is shown to the right, but Absolute Moves can also generate triangular profiles. The command's Target Position can be in the range of $\pm 1,073,741,823$ counts. The move will be clockwise if the Target Position is greater than the Current Position and counter-clockwise if the Target Position is less than the Current Position.

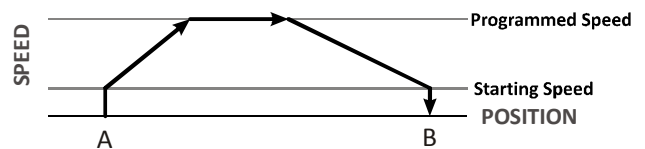


Figure R2.7 Absolute Move

- Note** ➤
- 1) The Motor Position must be valid before you can use an Absolute Move. The Motor Position becomes valid when you preset the position or home the machine. See the reference chapter 5: [Homing the IANF2\(E\)](#), which starts on page 37, for information on homing the machine.
 - 2) Absolute Moves allow you to move your machine without having to calculate relative positions. If you are controlling a rotary table, you can drive the table to any angle without having to calculate the distance to travel. For example an Absolute Move to 180° will move the table to the correct position regardless of where the move starts from.

Controlled Stop Conditions

- The move completes without error.
- The Hold Move control bit in the command data from the FC6A controller makes a 0→1 transition. This transition will bring the axis into its Hold State. The use of the Hold Move bit, the Resume Move bit, and the External Input is further explained in the *Controlling Moves In Progress* section starting on page 25.
- The External Input for the axis makes an inactive-to-active transition. This transition will bring the axis into its Hold State, with the axis acting as if the Hold Move control bit in the command data from the FC6A controller made a 0→1 transition. The use of the External Input, the Hold Move bit, and the Resume Move bit is further explained in the *Controlling Moves In Progress* section starting on page 25.

If the External Input is active when an Absolute Move is started, the IANF2(E) will output one step in the specified direction before entering the Hold State.

Immediate Stop Conditions

- The Immediate Stop bit makes a 0→1 transition in the command data from the FC6A controller.
- An inactive-to-active transition on an input configured as an E-Stop Input.
- A CW/CCW Limit Switch is reached.

Note ➤ An Input_Error bit is latched in the Network Input Data when an E-Stop or Limit Switch is activated. The next move command will clear this bit automatically if none of the inputs are active when the command is issued.

Jog CW/CCW Moves

Jog Moves move in the programmed direction as long as the command is active. Two commands are available. The Jog CW Move will increase the motor position count while the Jog CCW Move will decrease the motor position count. These commands are often used to give the operator manual control over the axis.

Jog Moves are also used when you are interested in controlling the speed of the shaft instead of its position. One such application is driving a conveyor belt. To accommodate these applications, the running speed, acceleration, and deceleration of the Jog Move can be changed *while the move is in progress*.

The CW Limit and CCW Limit inputs behave differently for Jog CW/CCW Moves and Registration Moves CW/CCW than all other move types. Like all moves, activating a limit that is the same as the direction of travel, for example activating the CW Limit during a Jog CW Move, will bring the move to an Immediate Stop. Unlike the other moves, activating a limit that is opposite to the direction of travel, for example activating the CCW Limit during a Jog CW Move, has no effect. This allows you to jog off of an activated limit switch.

As shown below, a Jog Moves begins at the programmed Starting Speed, accelerates at the programmed rate to the Programmed Speed and continues until a stop condition occurs. If it is a *Controlled Stop Condition*, the IANF2(E) will decelerate the pulse train to the Starting Speed and stop without losing position. If it is an *Immediate Stop Condition*, the pulse train stops immediately and the position becomes invalid.

It is possible to change the speed of a Jog Move without stopping the motion if the initial value of the Programmed Speed parameter is greater than the configured Starting Speed. The Programmed Speed, Acceleration, and Deceleration parameters can be changed during a Jog Move. When the Programmed Speed is changed, the motor will accelerate or decelerate to the new Programmed Speed using the new accelerate/decelerate parameter values.

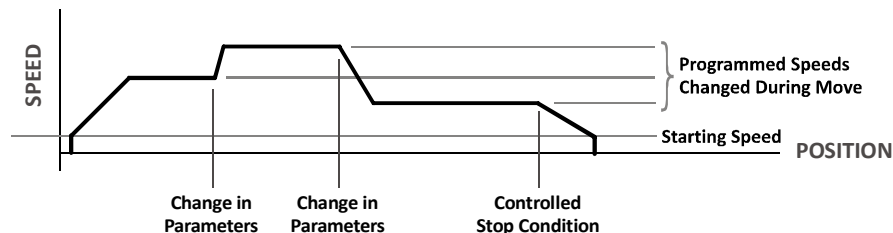


Figure R2.8 Jog Move

If the initial Programmed Speed parameter is less than the configured Starting Speed, the IANF2(E) will output pulses at the programmed rate. Attempting to change the Programmed Speed, Acceleration, or Deceleration parameters will result in an error.

If the initial Programmed Speed parameter equals zero, the IANF2(E) will output one pulse in the specified direction.

Controlled Stop Conditions

- The Jog Move Command bit is reset to "0".
- The External Input for the axis makes an inactive-to-active transition. This transition will bring the jog to a stop.

Immediate Stop Conditions

- The Immediate Stop bit makes a 0→1 transition in the command data from the FC6A controller.
- The Hold_Move bit makes a 0→1 transition in the command data from the FC6A controller. This transition will result in the Command_Error bit being set in the Network Input Data.
- An inactive-to-active transition on an input configured as an E-Stop Input.
- A CW/CCW Limit Switch is reached.

Note ➤ 1) An Input_Error bit is latched in the Network Input Data when an E-Stop or Limit Switch is activated. The next move command will clear this bit automatically if none of the inputs are active when the command is issued.

2) Note that it is possible to *start* a move while a CW or CCW Limit Switch is active as long as the direction of travel is *opposite* that of the activated Limit Switch. For example, it is possible to start a Jog CW Move while the CCW Limit Switch is active

Registration Move CW/CCW

Similar to a Jog Move, a Registration Move will travel in the programmed direction as long as the command is active. A Registration Move CW will increase the motor position count while a Registration Move CCW will decrease the motor position count. When the command terminates under Controlled Stop conditions, the IANF2(E) will output a programmed number of steps as part of bringing the move to a stop. Note that all position values programmed with a Registration Move are relative values, not absolute machine positions.

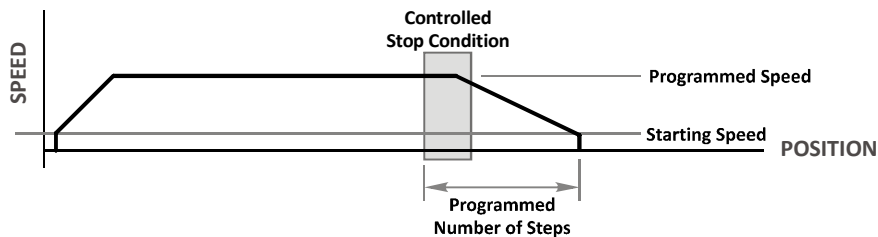


Figure R2.9 Registration Move

Note ➤ If the Programmed Number of Steps are less than the number of steps needed to bring the axis to a stop based on the Programmed Speed and Deceleration values set with the command, the IANF2(E) will issue an error message.

As with the Jog Moves, activating a limit that is opposite of the commanded direction has an effect. For example, activating the CCW Limit Switch during a Registration Move CW.

An additional feature of the Registration Moves is the ability to program the IANF2(E) to ignore the Controlled Stop conditions until a minimum number of steps have occurred. This value is programmed through the Minimum Registration Move Distance parameter, which is set when you command the Registration Move. The figure below shows how the Minimum Registration Move Distance parameter affects when the Stop Condition is applied to the move. As shown in the second diagram, Controlled Stop conditions are level triggered, not edge triggered. If a Controlled Stop Condition occurs before the Minimum Registration Move Distance is reached and the condition remains active, the move will begin its controlled stop once the Minimum Registration Move Distance is reached.

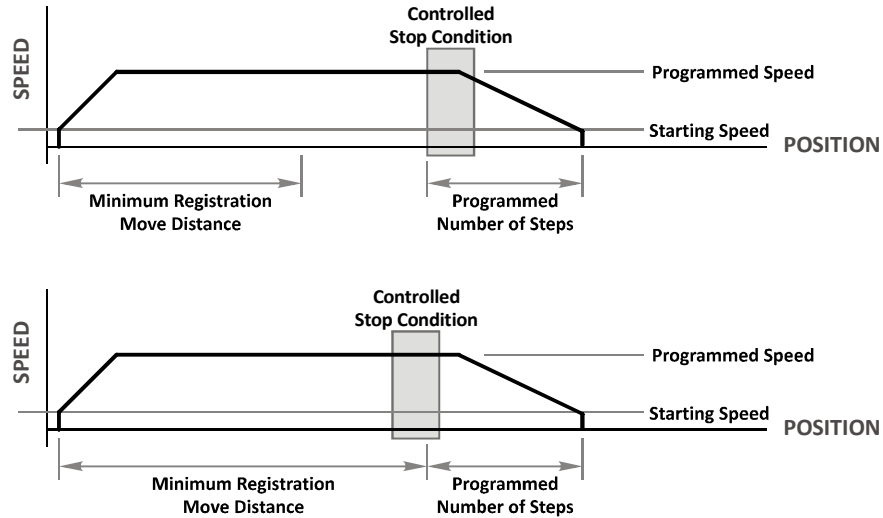


Figure R2.10 Minimum Registration Move Distance

Controlled Stop Conditions

- The Registration Move Command bit is reset to "0".
- An inactive-to-active transition on the External Input.

Note ➤ Starting a Registration Move with the External Input in its active state will result in an error if the *Minimum Registration Distance* equals zero, or a move of (*Minimum Registration Distance* + *Programmed Number of Steps*) if the *Minimum Registration Distance* is greater than zero.

Immediate Stop Conditions

- The Immediate Stop bit makes a 0→1 transition in the command data from the FC6A controller.
- The Hold_Move bit makes a 0→1 transition in the command data from the FC6A controller. This transition will result in the Command_Error bit being set in the Network Input Data.
- An inactive-to-active transition on an input configured as an E-Stop Input.
- A CW/CWW Limit Switch is reached.

Note ➤ 1) An Input_Error bit is latched in the Network Input Data when an E-Stop or Limit Switch is activated. The next move command will clear this bit automatically if none of the inputs are active when the command is issued.
 2) Note that it is possible to *start* a move while a CW or CCW Limit Switch is active as long as the direction of travel is *opposite* that of the activated Limit Switch. For example, it is possible to start a Jog CW Move while the CCW Limit Switch is active

Controlling Moves In Progress

The IANF2(E) has the ability to place a running move on hold. The module can later resume the move if an error did not occur while the move was in its Hold state. One potential application for this feature is bringing a move to a controlled stop when your controller senses an end-of-stock condition. The move can be put in its Hold state until the stock is replenished and then the move can be resumed.

Note that you do not have to resume a move once it has been placed in its Hold state. You can place a move in its Hold state to prematurely end the move with a controlled stop and issue any type of new move from the stopped position.

The figure below shows a profile of a move that is placed in its Hold state and later resumed.

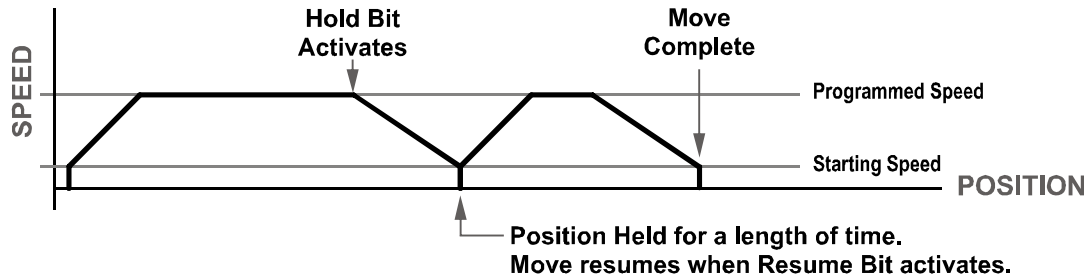


Figure R2.11 Hold/Resume a Move Profile

Find Home Moves

Issuing a Hold command while a Home CW or Home CCW command is active will result in the IANF2(E) issuing a Command Error and the axis will be brought to an immediate stop.

Jog Moves

Issuing a Hold command while a Jog CW or Jog CCW command is active will result in the IANF2(E) issuing a Command Error and the axis will be brought to an immediate stop.

Registration Moves

Issuing a Hold command while a Registration Move CW or Registration Move CCW command is active will result in the IANF2(E) issuing a Command Error and the axis will be brought to an immediate stop.

Absolute and Relative Moves

Absolute and Relative Moves can be placed in a Hold state and resumed if error conditions, such as programming errors, have not occurred. New Acceleration, Deceleration, and Programmed Speed parameters can be written to the IANF2(E) while these moves are in their hold states. If the parameters are accepted without error, the move can be resumed and it will use the new parameter values. Note that a change to the Target Position is ignored.

Encoder Follower Move

An encoder is typically used in an open-loop system for position verification. The encoder is mounted to the back of the controlled motor and the encoder position is read after a move to verify that the motor moved as expected.

When an axis of the IANF2(E) is used in Encoder Follower mode, the encoder is not mounted on the controlled motor. Instead, the encoder is typically mounted on a second motor, but it can be mounted anywhere, including on something as simple as a hand crank. While in this mode, the axis will output motor control pulses in response to pulses on the encoder inputs.

Two commands are available. The Encoder Follower CW Move will output clockwise pulses when the encoder count is increasing and the Encoder Follower CCW Move will output counter-clockwise pulses when the encoder count is increasing. The IANF2(E) considers the encoder count increasing when the encoder's A signal is leading the B signal.

The External Input is ignored when the axis is in Encoder Follower mode.

This mode is sometimes referred to as *electronic gearing*, because the motor will change position in response to a change in position of the encoder.

Multiplier and Divisor

The IANF2(E) has two parameters that allow you to set any pulse ratio you want between the encoder and the motor. These two parameters are the Multiplier and Divisor. When their ratio equals one, the IANF2(E) will output one pulse for every position change it sees on the encoder inputs.

Note ➤ As with all encoder functions on the IANF2(E), X4 decoding is used when updating the encoder count register. This fact must be taken into account when calculating the appropriate Multiplier and Divisor values.

The Multiplier and Divisor parameter each have a range of 1 to 32,768. It is possible to enter a ratio greater than one when programming these values. The IANF2(E) will output multiple steps per encoder count.

Note ➤ The maximum encoder input frequency is 250 kHz, and the maximum motor output frequency is 1 MHz.

It is possible for your application to mathematically exceed this 1 MHz maximum output frequency when the Multiplier/Divisor ratio is greater than one. For example, assume your Multiplier/Divisor ratio is 3, and your encoder input frequency is 125 kHz. Ideally, your motor output frequency would be: $125 \text{ kHz} \times 4\{\text{decoding}\} \times 3\{\text{ratio}\} = 1.5\text{MHz}$. As this exceeds the 1 MHz limit, the IANF2(E) will output pulses at its 1MHz rate until motor position catches up with the encoder position.

Acceleration/Deceleration

The Linear Acceleration type is the only one available when using Encoder Follower Moves. Unlike other moves, the Acceleration and Deceleration values can be set to zero with this move type. If they are zero, the axis will immediately match the encoder speed when a move begins. If this stalls the motor, set the Acceleration and Deceleration parameters to their maximum value of 2000 steps/millisecond/second. If the motor still stalls, decrease the Acceleration and Deceleration values until the motor runs correctly.

Note ➤ 1) The IANF2(E) will always output the correct number of steps when operating in this mode. However, the use of the Acceleration and Deceleration parameters, especially low values, can cause the motor's motion to lag behind the encoder's motion.
2) If the Acceleration and Deceleration parameters are set to zero, electrical noise on the encoder inputs may cause the motor to oscillate back and forth by ± 1 step when there is no encoder motion.

How It Works

The IANF2(E) always uses 4X decoding when counting pulses from the encoder. If you set both of your Multiplier and Divisors to 1 and set the motor steps/turn on your motor's driver to four times the number of encoder lines, then the motor will complete one rotation for every rotation of the encoder's shaft.

Once placed in Encoder Follower mode, the IANF2(E) monitors the Jog Move command bits in the output registers assigned to the module. When either of these bits are set, the encoder inputs are monitored for a change in position. When a change is sensed, the IANF2(E) will begin to output pulses within 50 microseconds.

Controlled Stop Conditions

- The encoder stops moving.
- Both of the Jog Move command bits equal zero.
- Encoder Follower moves cannot be brought to a controlled stop by using the Hold Move control bit in the command data from the FC6A controller.

Immediate Stop Conditions

- The Immediate Stop bit makes a 0 → 1 transition in the command data from the FC6A controller.
- A positive transition on an input configured as an E-Stop Input.
- A CW or CWW Limit Switch is reached.

Advanced Ratio Control

The Multiplier and Divisor values give you a great deal of control over the ratio of motor turns per encoder turn, but you can achieve even finer control if your motor's driver allows you to set the motor steps per turn.

The Z pulse is not used to correct the encoder position once per turn, so you can actually program the Motor Steps/Turn to any value you want within its valid range. For example, if your encoder's resolution is 4,000 counts per turn (a 1,000 line encoder) and you set the motor steps per turn on the motor's driver to 10,000, you will have built a 2.5:1 gear down into your system before applying the IANF2(E) Multiplier and Divisors. (Two and one half rotations of the encoder = 10,000 counts = 1 motor rotation.)

REFERENCE 3: CALCULATING MOVE PROFILES

Introduction

This reference is for customers that must program very precise profiles. Understanding this section is not necessary before programming the IANF2(E) and it can be considered optional.

Units of Measure

The equations in this reference use a unit of measure for acceleration and deceleration of steps/second/second (steps/second²). However, when programming the IANF2(E), all acceleration and deceleration values must be programmed in the unit of measure of steps/second/millisecond.

- To convert from steps/second² to steps/millisecond/second, divide the value by 1000. This must be done when converting from a value used in the equations to a value programmed into the IANF2(E).
- To convert from steps/millisecond/second to steps/second², multiply the value by 1000. This must be done when converting from the value programmed into the IANF2(E) to the value used in the equations.

Linear Acceleration Equations

When you choose to use the linear acceleration type, the speed of the move will increase linearly towards the Programmed Speed. This is the fastest form of acceleration, resulting in the fastest move between two points at its programmed speed. For the smoothest transition from the starting speed, the starting speed should be equal to the square root of the acceleration in steps/sec². For example, if the choose acceleration is 20,000 steps/sec², the smoothest transition occurs when the starting speed is 141. ($141^2 \approx 20,000$)

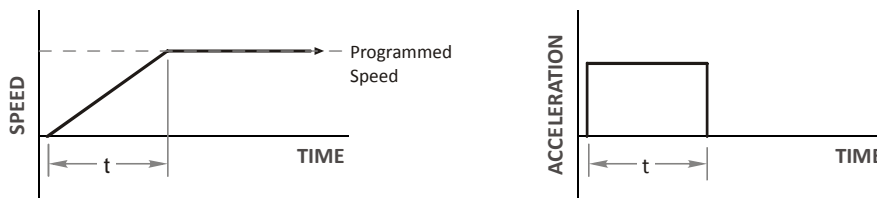


Figure R3.1 Constant Acceleration Curves

Variable Definitions

The following variables are used in these equations:

- V_S = Configured Starting Speed of the move
- V_P = Programmed Speed of the move
- a = Acceleration value. Must be in the units of steps/second²
- d = Deceleration value. Must be in the units of steps/second²
- T_A or T_D = Time needed to complete the acceleration or deceleration phase of the move
- D_A or D_D = Number of Steps needed to complete the acceleration or deceleration phase of the move

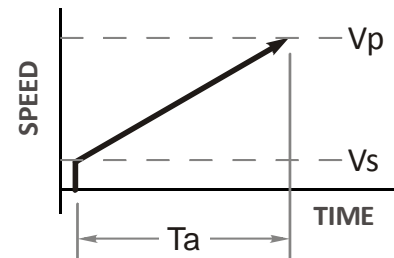


Figure R3.1 gives the equations to calculate Time, Distance, and Acceleration values for a constant acceleration move.

Acceleration Type	T_A or T_D (Time to Accelerate or Decelerate)	D_A or D_D (Distance to Accelerate or Decelerate)	a (Average Acceleration)
Linear	$T_A = (V_P - V_S)/a$ $T_D = (V_P - V_S)/d$	$D_A = T_A * (V_P + V_S)/2$ $D_D = T_D * (V_P + V_S)/2$	$a = (V_P^2 - V_S^2)/2D_A$

Table R3.1 Acceleration Equations

If the sum of the D_A and D_D values of the move is *less than* the total number of steps in the move, your move will have a Trapezoidal profile.

If the sum of the D_A and D_D values of the move is *equal to* the total number of steps in the move, your move will have a Triangular profile and your move will reach the Programmed Speed before it begins to decelerate.

If the sum of the D_A and D_D values of the move is *greater than* the total number of steps in the move, your move will have a Triangular profile and it *will not* reach the Programmed Speed before it begins to decelerate.

As an example, let's assume the values in table R3.2 for a move profile.

Name	Value	IANF2(E) Parameter Values
Acceleration (a)	20,000 steps/sec ²	20
Deceleration (d)	25,000 steps/sec ²	25
Starting Speed (V _S)	141 steps/sec	141
Programmed Speed (V _P)	100,000 steps/sec	100,000

Table R3.2 Sample Values

From table R3.1:

$$\text{Time to accelerate: } T_A = (V_P - V_S)/a = (100,000 - 141)/20,000 = 4.993 \text{ seconds}$$

$$\text{Time to decelerate: } T_D = (V_P - V_S)/d = (100,000 - 141)/25,000 = 3.994 \text{ seconds}$$

$$\text{Distance to Accelerate: } D_A = T_A * (V_P + V_S)/2 = 4.993 * (100,000 + 141)/2 = 250,002 \text{ steps}$$

$$\text{Distance to Decelerate: } D_D = T_D * (V_P + V_S)/2 = 3.994 * (100,000 + 141)/2 = 199,982 \text{ steps}$$

$$\text{Total Distance needed to accelerate and decelerate: } 250,002 + 199,982 = 449,984 \text{ steps}$$

If a move with the above acceleration, deceleration, starting speed, and programmed speed has a length greater than 449,984 steps, the IANF2(E) will generate a Trapezoidal profile. If the move is equal to 449,984 steps, the IANF2(E) will generate a Triangular profile and the unit will output one pulse at the programmed speed. If the move is less than 449,984 steps, the IANF2(E) will generate a Triangular profile and the programmed speed will not be reached.

In the case of a Triangular profile where the programmed speed is not reached, it is fairly easy to calculate the maximum speed (V_M) attained during the move. Because the move is always accelerating or decelerating, the total distance traveled is equal to the sum of D_A and D_D.

$$D_A = T_A * (V_M + V_S)/2 \text{ and } T_A = (V_M - V_S)/a. \text{ By substitution:}$$

$$D_A = (V_M - V_S)/a * (V_M + V_S)/2 = (V_M^2 - V_S^2)/2a. \text{ By the same method,}$$

$$D_D = (V_M^2 - V_S^2)/2d.$$

Therefore, total distance traveled =

$$D_A + D_D = (V_M^2 - V_S^2)/2a + (V_M^2 - V_S^2)/2d.$$

In the case where the acceleration and deceleration values are equal, this formula reduces to:

$$D_A + D_D = (V_M^2 - V_S^2)/a$$

Continuing the example from table R3.2, assume a total travel distance of 300,000 steps.

$$D_A + D_D = \frac{V_M^2 - V_S^2}{2a} + \frac{V_M^2 - V_S^2}{2d}$$

$$300,000 \text{ steps} = \frac{V_M^2 - 141^2}{2(20,000)} + \frac{V_M^2 - 141^2}{2(25,000)}$$

$$300,000 \text{ steps} = \frac{V_M^2 - 20,000}{40,000} + \frac{V_M^2 - 20,000}{50,000}$$

$$300,000 \text{ steps} = \frac{5}{5} \left(\frac{V_M^2 - 20,000}{40,000} \right) + \frac{4}{4} \left(\frac{V_M^2 - 20,000}{50,000} \right)$$

$$300,000 \text{ steps} = \frac{5V_M^2 - 100,000}{200,000} + \frac{4V_M^2 - 80,000}{200,000}$$

$$300,000 (200,000) = 9V_M^2 - 180,000$$

$$\frac{60,000.18 \times 10^6}{9} = V_M^2$$

$$V_M = 81,650 \text{ steps/sec}$$

Once the maximum speed has been calculated, substitute this value into the time and distance formulas in table R3.1 to calculate time spent and distance traveled while accelerating and decelerating.

Total Time Equations

For Trapezoidal Profiles you must first determine the number of counts that you are running at the Programmed Speed. This value, (D_P below), is equal to your D_A and D_D values subtracted from your total travel. You can then calculate your total profile time, (T_T below), from the second equation.

$$D_P = (\text{Total Number of Steps}) - (D_A + D_D)$$

$$T_T = T_A + T_D + D_P/V_P$$

For Triangular Profiles, the total time of travel is simply:

$$T_T = T_A + T_D$$

S-Curve Acceleration Equations

S-Curve accelerations smoothly change the acceleration value applied during the move. This limits mechanical shocks to the system as the load accelerates. Triangular S-Curve accelerations vary the applied acceleration throughout the acceleration phase. Trapezoidal S-Curve accelerations have a period of linear acceleration in the middle of the acceleration phase.

When using variable accelerations, the starting speed does not have to be equal to the square root of the programmed acceleration value for the smoothest transition from stopped to accelerating. S-Curve accelerations provides smooth transitions at the beginning and end of the acceleration phase.

Triangular S-Curve Acceleration

The left half of figure R3.2 shows the speed profile of a move during its acceleration phase. The figure shows the triangular S-curve acceleration in red along with the equivalent constant acceleration in black. The equivalent constant acceleration is equal to the change in speed divided by the time it takes to achieve this change in speed.

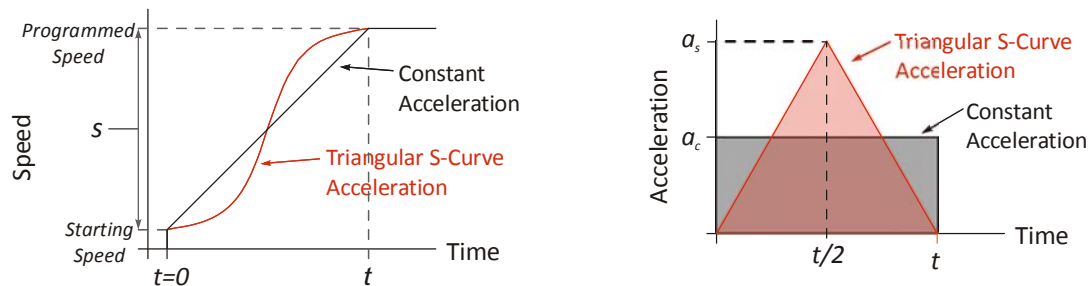


Figure R3.2 Triangular S-Curve Example

Speed is equal to acceleration multiplied by the time it is applied. The right side of figure R3.2 shows this graphically as the area of the gray rectangle. In order for the Triangular S-curve acceleration to reach the same speed in the same amount of time, the area of the triangle must equal the area of the square. The area of a triangle is one half of the base length multiplied by the height. Therefore:

$$a_c t = \frac{a_s t}{2} \quad \text{Area of rectangle} = \text{Area of triangle}$$

$$a_s = 2a_c$$

This means that a triangular S-curve acceleration profile requires twice the programmed maximum acceleration as a constant acceleration profile to achieve the same speed in the same amount of time.

The above examples assume that you can increase the programmed acceleration value to keep the acceleration time the same. If your constant acceleration value is the maximum your system will allow, then using S-curve accelerations will double the time needed to accelerate to your desired speed.

Trapezoidal S-Curve Acceleration

The left half of figure R3.3 shows the speed profile of a move during its acceleration phase. The figure shows the trapezoidal S-curve acceleration in red. In the IANF2(E), the period of constant acceleration is 50% of the acceleration phase. The equivalent constant acceleration is in black. The equivalent constant acceleration is equal to the change in speed divided by the time it takes to achieve the change in speed.

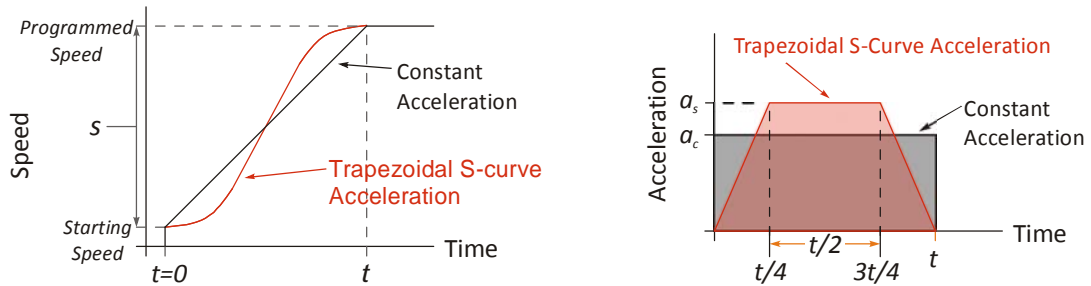


Figure R3.3 Move Profile Example

Speed is equal to acceleration multiplied by the time it is applied. The right side of figure R3.3 shows this graphically as the area of the gray rectangle. In order for the Trapezoidal S-curve acceleration to reach the same speed in the same amount of time, the area of the polygon must equal the area of the rectangle.

$$\frac{a_s t}{2} + \frac{a_s t}{4} = a_c t \text{ Area of polygon} = \text{Area of rectangle}$$

$$\frac{2a_s t}{4} + \frac{a_s t}{4} = a_c t$$

$$\frac{3a_s t}{4} = a_c t$$

$$a_s = \frac{4}{3}a_c$$

This means a trapezoidal S-curve acceleration profile requires its programmed acceleration value to be 4/3 that of the constant acceleration value used to achieve the same speed in the same amount of time.

The above examples assume that you can increase the programmed acceleration value to keep the time of the acceleration phase the same. If your constant acceleration value is the maximum your system will allow, then using S-curve accelerations will lengthen the time needed to accelerate to your desired speed by 133.3% (4/3).

REFERENCE 4: INTERPOLATED MOVE PROFILES

Introduction

This chapter introduces the three interpolated move types available with the IANF2(E). These moves coordinate the outputs of the two channels, allowing you to perform synchronized moves.

The X-Y Plane

Its easiest way to explain the interpolated move types is in terms of motion in the X-Y plane defined by the two axes. This plane is shown in figure R4.1.

- A move that results in increasing counts on the axis will cause CW pulses on the outputs of the axis. Likewise, a move that results in decreasing counts on the axis will cause CCW pulses on the outputs of the axis.
- The ± 2 billion+ counts at the ends of each axis represent the minimum and maximum values of the Current Position register for the axis, which is a signed 32 bit value. *These values are not hard limits.* If you are performing a relative move in a CW direction that exceeds 2,147,483,647 counts, the Current Position value will roll over to its maximum negative value.
- The colored square with limits of $\pm 1,073,741,823$ counts represents the limits of Absolute coordinates. All defined positions must be within these limits when programming an absolute move, but, in the case of circular interpolated moves, the move can travel beyond these limits while it is running.

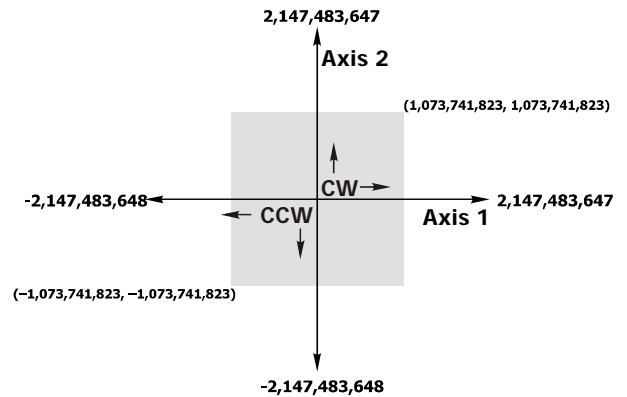


Figure R4.1 X-Y Plane

Linear Interpolated Moves

Conceptually, a Linear Interpolated Move is performed by travelling the shortest distance between two points on the X-Y plane.

The Start Point is the current position defined by the two axes. The End Point can be programmed with Relative or Absolute coordinates. In figure R4.2 on the right, relative coordinates are shown in parentheses.

Relative Linear Move

When programming a move with Relative coordinates, you program the number of steps, or offset, you want each axis to travel.

- When programming the End Point with Relative coordinates, the maximum offset is $\pm 1,073,741,823$ counts. Positive offsets will result in clockwise moves on the axis, while negative offsets result in counter-clockwise moves.
- The Current Motor Position register that is reported back to the PLC is a full 32 bit register. ($\pm 2,147,483,647$ counts) The only way to move beyond $\pm 1,073,741,823$ counts is with multiple relative moves.

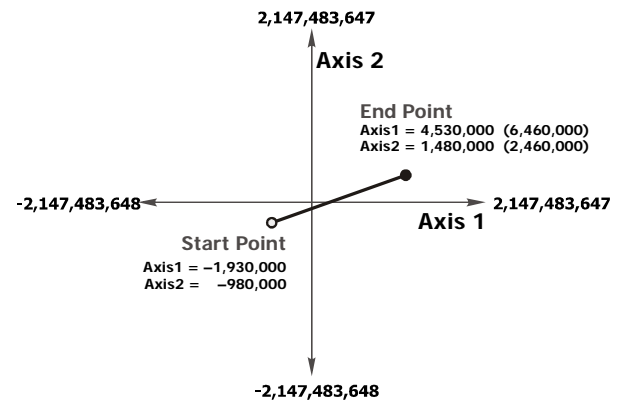


Figure R4.2 Linear Interpolated Move

Absolute Linear Move

Absolute coordinates treat the End Point as an actual position on the machine. Note that you must set the Home Position for both axes of the machine before you can run an Absolute Linear Move. (See the definition of *Home Position* on page 17.)

- The range of values for the End Point when it is treated as an actual position on the machine is $\pm 1,073,741,823$ counts. The move will be clockwise if the End Position coordinate is greater than the Current Position of the axis and negative if the End Position coordinate is less than the Current Position of the axis.
- The Start Point can be any position in the X-Y plane. It does not have to be within the $\pm 1,073,741,823$ count range.
- The Current Motor Position register that is reported back to the PLC is a full 32 bit register. ($\pm 2,147,483,647$ counts) However, you cannot move beyond $\pm 1,073,741,823$ counts with an Absolute Linear Move.

Required Parameters

Five parameters are required to define a Linear Interpolated Move:

- End Point X (Axis1) coordinate (Absolute or Relative)
- End Point Y (Axis2) coordinate (Absolute or Relative)
Must use same system as the X coordinate
- Interpolated Target Speed
- Interpolated Acceleration
- Interpolated Deceleration

Target Speed, Acceleration, and Deceleration define the vector for the path as a whole, not the individual axes, so they are programmed only once. As shown in figure R4.3 to the right, the IANF2(E) calculates the required component vector values for each axis before beginning the move.

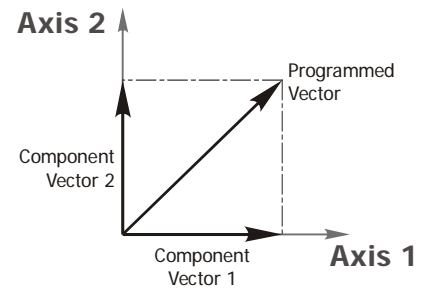


Figure R4.3 Programmed Vectors

Circular Interpolated Moves

Conceptually, a Circular Interpolated Move is performed by traveling between two points in the X-Y plane along an arc of a circle defined within the plane. As with Linear Interpolated Moves, the Start Point of every move is the current position of the two axes. The other points can be programmed with Absolute or Relative coordinates.

For non-interpolated, single axis moves, the terms “clockwise” and “counter-clockwise” refer to whether or not the Motor Position value reported back to the FC6A will increase or decrease as a result of the move. (Clockwise moves increase the Motor Position Value.) For Circular Interpolated Moves, the terms “clockwise” and “counter-clockwise” refer to the direction of travel when looking down at the X-Y plane. For example, figure R4.4 below shows a counter-clockwise move.

There are three methods to specify a Circular Interpolated Move.

Center Point Method

The Center Point method is shown in figure R4.4. In addition to the End Point, this method defines the Center Point of the circle. The IANF2(E) verifies that the length from the Center Point to the Start Point is equal to the length from the Center Point to the End Point before running the profile. **Both lengths are radii of the circle and must be equal.**

Even though these three points completely define the circle, they cannot define the direction of travel along the circle to move from the Start Point to the End Point. Because of this, the IANF2(E) has two commands for use with the Center Point method. One command causes CW motion along the circle path while the other causes CCW motion. Note that CW and CCW motion in this case refers to the direction of travel in the X-Y plane as shown in the figure above. It does not refer to the state of the outputs during the move. Depending on the size and location of the circular path, each axis may output both CW and CCW pulses during the move.

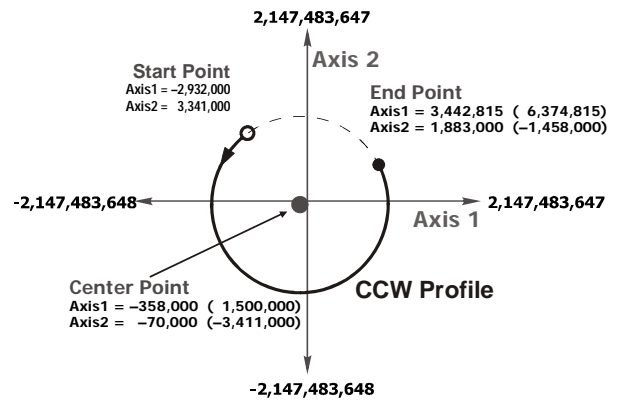


Figure R4.4 Center Point Circular Move

The Center Point and End Point can be specified with Relative or Absolute coordinates and the two points must use the same coordinate system.

- When programming the End Point and Center Point with Relative coordinates, the maximum offsets are ±1,073,741,823 counts from the Start Point.
- When programming the End Point and Center Point with Absolute coordinates, these positions must be in the range of ±1,073,741,823 counts. The Start Point can be any position in the X-Y plane. It does not have to be within the ±1,073,741,823 count range.
- The IANF2(E) calculates the radius value and it cannot exceed 1,073,741,823 counts.

Note ➤ This is the only method that allows you to set the End Point equal to the Start Point and travel along the entire circular path in the X-Y plane. All other methods only allow you to travel over an arc of the defined circle.

Circular Interpolated Moves (continued)

Radius Method

The Radius method is shown in figure R4.5. In addition to the End Point, this method defines radius of the circular path. In addition to the Start Point, these two pieces of information actually define two circles in the plane, so an additional piece of information is used to define the move path. This piece of additional information is the sign of the radius value.

- If the radius is positive, the move will travel the shortest arc between the two points. If the radius is negative, the moves will travel the longest arc between the two points.

In order to determine the direction of travel, clockwise or counter-clockwise, the IANF2(E) has two commands for use with the Radius method. One command causes CW motion along the circle path while the other causes CCW motion. Note that CW and CCW motion in this case refers to the direction of travel in the X-Y plane as shown in the figure R4.5. It does not refer to the state of the outputs during the move.

The bottom half of figure R4.5 shows the four available moves based on the sign of the radius value and the move direction.

- **Path A:** CW move, Negative radius
- **Path B:** CW move, Positive radius
- **Path C:** CCW move, Positive radius
- **Path D:** CCW move, Negative radius

The End Point can be specified with Relative or Absolute coordinates.

- When programming the End Point with Relative coordinates, the maximum X and Y offsets are $\pm 1,073,741,823$ counts from the Start Point.
- When programming the End Point with Absolute coordinates, the X and Y positions must be in the range of $\pm 1,073,741,823$ counts. The Start Point can be any position in the X-Y plane. It does not have to be within the $\pm 1,073,741,823$ count range.
- The End Point cannot have the same coordinates as the Start Point. Therefore, a move programmed by this method cannot travel along the entire circular path in the X-Y plane. If you need your move to travel the entire circular path, then you must program the move using the [Center Point Method](#) as described on page 32.
- The maximum value for the Radius parameter is $\pm 1,073,741,823$ counts.

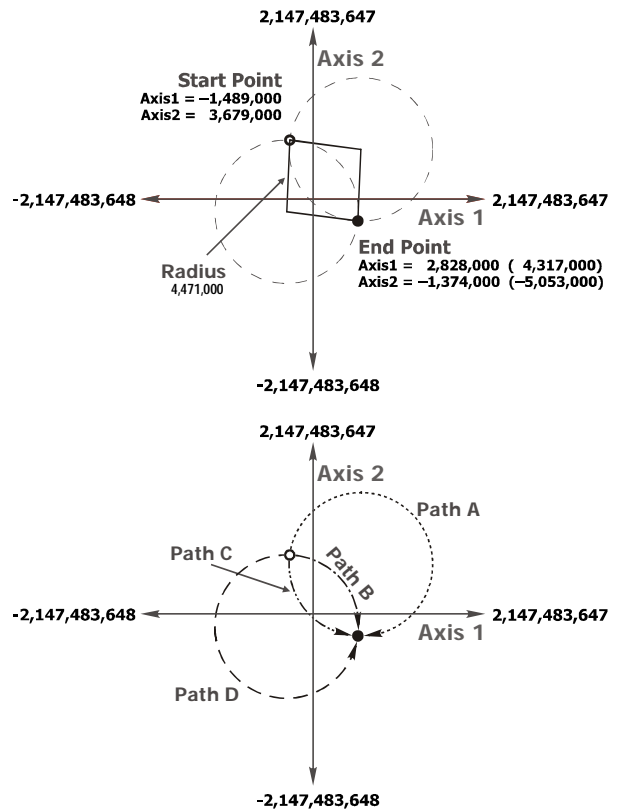


Figure R4.5 Radius Circular Move

Via Point Method

The Via Point Method defines a third point on the circular path that the move will pass through while travelling from the Start Point to the End Point. This method is illustrated in figure R4.6.

This method has only one command associated with it because the IANF2(E) can determine the direction of travel for the move with the three given points.

The Via Point and End Point can be specified with Absolute or Relative coordinates and the two points must use the same coordinate system.

- When programming the End Point and Via Point with Relative coordinates, the maximum X and Y offsets are $\pm 1,073,741,823$ counts from the Start Point.
- When programming the End Point and Via Point with Absolute coordinates, the X and Y positions must be in the range of $\pm 1,073,741,823$ counts. The Start Point can be any position in the X-Y plane. It does not have to be within the $\pm 1,073,741,823$ count range.
- The maximum value for the Radius parameter is 2,147,483,647 counts.
- Note that the Start Point, End Point, and Via Point must all have different coordinates and they cannot be collinear. Therefore, a move programmed by this method cannot travel along the entire circular path in the X-Y plane. If you need your move to travel the entire circular path, then you must program the move using the *Center Point Method* as described on page 32.

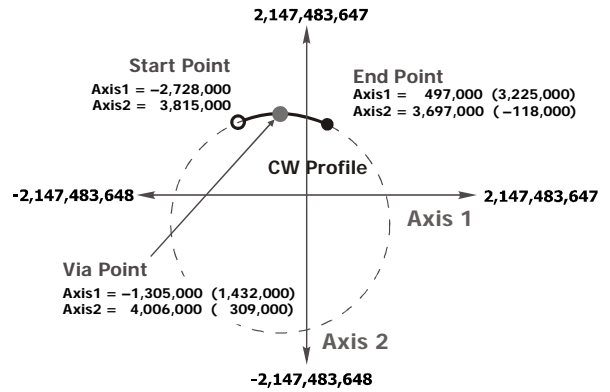


Figure R4.6 Via Point Circular Move

Controlled and Immediate Stops

Like most of the single axis moves described in reference chapter 2, Interpolated Moves can be brought to a Controlled Stop or Immediate Stop.

- **Controlled Stop:** The move immediately begins decelerating at its programmed deceleration value until it reaches the configured Starting Speed. The move stops at this point. The motor position values are still considered valid after a Controlled Stop and the machine does not need to be homed before Absolute Moves can be run again.
- **Immediate Stop:** The move immediately stops outputting pulses regardless of the present pulse frequency. Because it is possible for the inertia of the load attached to the motors to pull them beyond their stopping points, the motor position values are considered invalid after an Immediate Stop and the machine must be homed before Absolute Moves can be run again.

Controlled Stop Conditions

- The Hold Move control bit in the command data from the FC6A controller makes a 0→1 transition. This transition will bring the axis into its Hold State. This command should be issued on axis 1.
- The External Input for either axis makes an inactive-to-active transition. This transition will bring the axis into its Hold State.

If the External Input is active when a Relative Move is started, the IANF2(E) will output one step in the specified direction before entering the Hold State.

Immediate Stop Conditions

- The Immediate Stop bit makes a 0→1 transition in the command data from the FC6A controller.
- An inactive-to-active transition on an input configured as an E-Stop Input.
- A CW/CWW Limit Switch is reached.

Handling Rounding Errors

When you enter coordinates for Interpolated Moves, you enter integer values. The IANF2(E) then calculates the path it needs to take before actually starting the move. For Linear Interpolated Moves, the IANF2(E) will always be able to calculate a move profile as long as the end point is within its valid range. This is not the case for Circular Interpolated Moves.

Regardless of the method you use to define a Circular Interpolated Move, the IANF2(E) always calculates the center point of the circle as well as the radius from the center point to the start point and the radius from the center point to the end point. If these radii are not equal, then the IANF2(E) generates an error message instead of performing the move.

More likely than not, the two radii will be real values with fractional components. These two values are rounded off by the IANF2(E) to integers before being compared. The IANF2(E) will round off these values as you would expect, all values with a fractional component greater than or equal to 0.5 are rounded up, all values with a fractional component less than 0.5 are rounded down.

Figure R4.7 is an example of this problem. The radius from the Center Point and Start Point is approximately 1,828.6, which is rounded to the integer 1,829 by the IANF2(E). The radius from the Center Point and End Point is approximately 1,828.2, which is rounded to the integer 1,828. Because these radii differ by one count, the IANF2(E) will refuse to run the move.

You have two options to correct this problem. You can adjust the Center Point location or you can adjust the End Point location. The solution you choose depends on which point location is more important for you to keep. In the example, the End Point was adjusted by increasing the axis 1 position by one count. This is the simplest solution because adjusting the Center Point would require you to recalculate the Center Point to Start Point radius because it will also change.

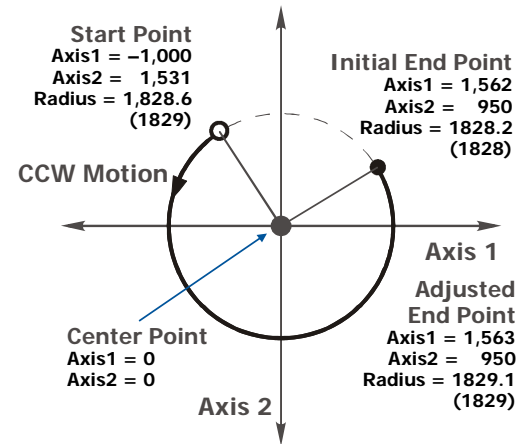


Figure R4.7 Round Off Error Example

Notes

REFERENCE 5: HOMING THE IANF2(E)

Introduction

This chapter explains the various ways of homing an axis on an IANF2(E). Inputs used to home the module are introduced and diagrams that show how the module responds to a homing command are given.

Definition of Home Position

The Home Position is any position on your machine that you can sense and stop at. Once at the Home Position, the motor position register on the IANF2(E) must be set to an appropriate value. If you use the module's *Home CW* and *Home CCW* commands, the motor position register will automatically be reset to zero once the home position is reached. The Encoder Position register will also be reset to zero if the quadrature encoder is enabled for the axis.

Note ➤ Defining a Home Position is completely optional. Some applications, such as those that use a servo or stepper for speed control, don't require position data at all.

With the exception of Absolute Moves, the IANF2(E) can still perform all of its move commands if the Home Position is not defined. Both axes must be homed before you can run Absolute Interpolated Moves.

Preset Position

One of the ways to define the Home Position is to issue the Preset Position command to the IANF2(E) axis. Before doing this, you will need a way of sensing position outside of the IANF2(E) module. The machine position data must be brought into the host controller, the correct preset value calculated, and this value written to the IANF2(E) axis with the Preset Position command. A separate Preset Encoder command exists that allows you to set the encoder position register to any value within its range. The range of values for motor position and the encoder position is -2,147,483,648 to 2,147,483,647.

Home CW/CCW Commands

The other choice is to use the module's Home CW/CCW commands to order the IANF2(E) to find the Home Position based on sensors brought into the unit. The Home CW command begins searching by outputting CW pulses to the motor's driver and ends when the home sensor triggers while the IANF2(E) is outputting CW pulses at the Starting Speed. The Home CCW command operates in the same way but starts and ends with CCW pulses.

Homing Inputs

Five inputs can be used when homing the module. Four are physical inputs attached to the module and one is a bit in the FC6A output data words.

Physical Inputs

- **Home Input:** This input is used in one of two ways: 1) This input is used to define the actual home position of the machine. 2) The input is used as a home proximity input when using the encoder marker pulse to home the machine.
- **Encoder Marker (Z) Pulse:** If you configure the IANF2(E) to use an encoder, you have the option of using the encoder's marker pulse to home the IANF2(E).
- **CW Limit Switch Input:** This input is used to prevent overtravel in the clockwise direction.
- **CCW Limit Switch Input:** This input is used to prevent overtravel in the counter-clockwise direction.

Backplane Inputs

- **Backplane_Proximity_Bit:** The IANF2(E) can be configured to ignore changes on the physical homing input until the Backplane_Proximity_Bit makes a 0→1 transition. The IANF2(E) will home on the next inactive-to-active change on the physical input once this transition occurs. You must program your FC6A to control the state of this bit.

Note ➤ This bit is not used by default, and must be activated when you configure the IANF2(E) before it can be used. If you decide to activate this bit when you configure the unit and then never set the Backplane_Proximity_Bit to a "1" while homing the IANF2(E), motion will not stop when the physical homing input is reached.

Homing Configurations

The IANF2(E) axis must be correctly configured before one of the homing commands will be accepted. One of the following must be part of the axis configuration before you can run the homing commands.

- 1) Enable the Home Input. Configure the homing operation to use the Home Input.
- 2) Enable the Home Input as well as the Backplane_Proximity_Bit. Configure the homing operation to use the Home Input.
- 3) Enable the Home Input as well as the Encoder Inputs. Configure the homing operation to use the Marker Pulse. (In this configuration, the Home Input acts as a hardware home proximity input.)
- 4) Enable the Encoder Inputs as well as the Backplane_Proximity_Bit. Configure the homing operation to use the Marker Pulse.

Note ▶ A homing operation can occur without configuring the axis to use the CW Limit or CCW Limit inputs. If you choose to configure the axis in this way, the axis has no way to automatically prevent overtravel during a homing operation. In linear applications, you must prevent overtravel by some external means, or ensure that the homing command is issued in the direction that will result in reaching the homing input directly.

In addition to the above settings, the Homing Timeout parameter will also be set. This parameter sets a maximum number of seconds the IANF2(E) axis will attempt to find the Home Position before it fails with an error. The range is 0 to 300 seconds, with a value of zero disabling the timeout feature.

Homing Profiles

Note ▶ The Home CW command is used in all of these examples. The Home CCW command will generate the same profiles in the opposite direction.

Home Input Only Profile

Figure R5.1 below shows the move profile generated by a Home CW command when you use the Home Input without the Backplane_Proximity_Bit.

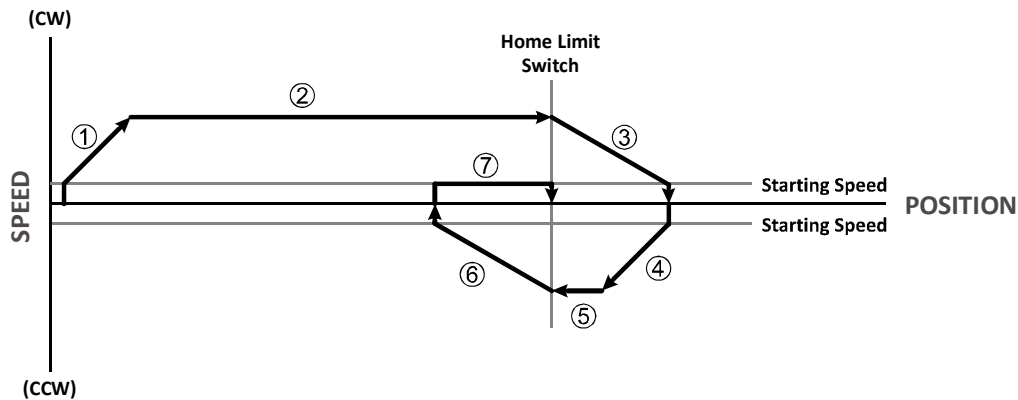


Figure R5.1 Home Input Profile

- 1) Acceleration from the configured Starting Speed to the Programmed Speed
- 2) Run at the Programmed Speed until the Home Input activates
- 3) Deceleration to the Starting Speed and stop, followed by a two second delay.
- 4) Acceleration to the Programmed Speed opposite to the requested direction.
- 5) Run opposite the requested direction until the Home Input transitions from Active to Inactive
- 6) Deceleration to the Starting Speed and stop, followed by a two second delay.
- 7) Return to the Home Input at the configured Starting Speed. Stop when the Home Input transitions from inactive to active.

Note ▶ If the Home Input is active when the command is issued, the move profile begins at step 5 above.

Profile with Proximity Input

Figure R5.2 below shows the move profile generated by a +Find Home command when you use:

- the Home Input with Backplane_Proximity_Bit
- the Marker Pulse home with Home Input as proximity sensor
- the Marker Pulse home with Backplane_Proximity_Bit

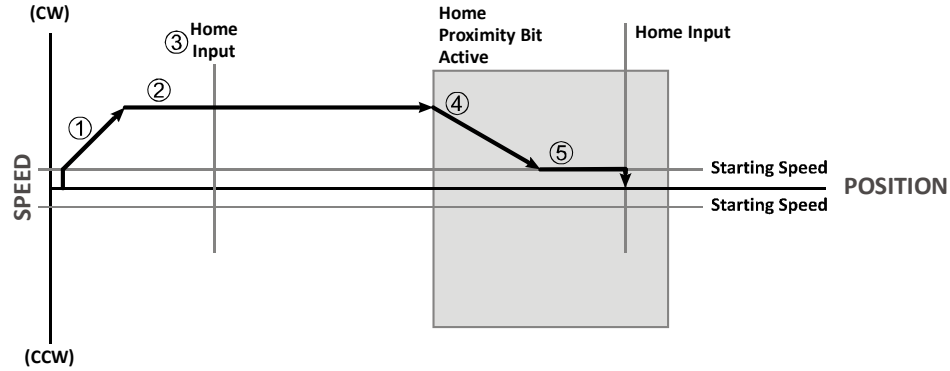


Figure R5.2 Homing with Proximity

- 1) Acceleration from the configured Starting Speed to the Programmed Speed
- 2) Run at the Programmed Speed
- 3) Ignores homing input, (Home Input or Marker Pulse) because the proximity input has not made an inactive to active transition.
- 4) Deceleration towards the Starting Speed when the proximity input (Backplane bit or Home Input) transitions from its inactive to active state. The axis will stop as soon as the Home Input becomes active.
- 5) The Starting Speed is the minimum speed the profile will run at. If the axis decelerates to the Starting Speed before reaching the Home Input, it will continue at this speed.

Note ➤ Figure R5.2 shows the Backplane_Proximity_Bit staying active until the IANF2(E) reaches its home position. This is valid, but does not have to occur. As stated in step 4, the IANF2(E) starts to hunt for the home position as soon as the Backplane_Proximity_Bit makes a 0→1 transition.

Profile with Overtravel Limit

Figure R5.3 below shows the move profile generated by a +Find Home command when you use:

- CW Overtravel Limit
- Home Input without the Backplane_Proximity_Bit

The profile is generated when you encounter an overtravel limit in the direction of travel. (In this example, hitting the CW limit while traveling in the CW direction.) Hitting the overtravel limit associated with travel in the opposite direction is an Immediate Stop condition. The axis will stop all motion and issue a *Home Invalid* error to the FC6A.

The IANF2(E) will stop the axis with an error if both overtravel limits are activated while the IANF2(E) is trying to find the home position.

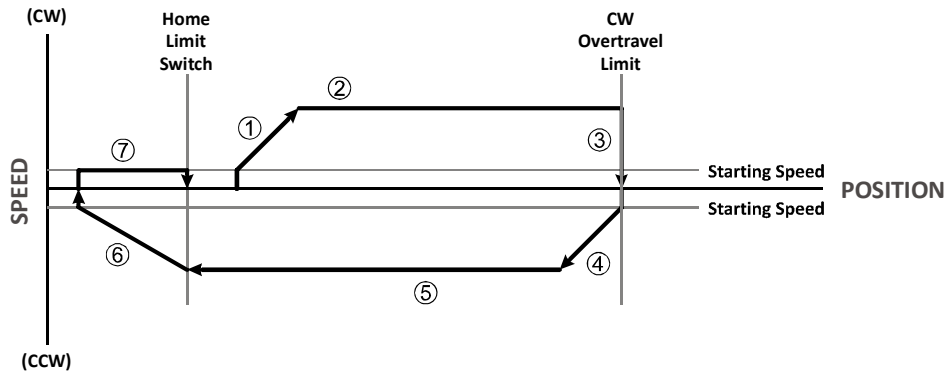


Figure R5.3 Profile with Overtravel Limit

- 1) Acceleration from the configured Starting Speed to the Programmed Speed.
- 2) Run at the Programmed Speed.
- 3) Hit CW Limit and immediately stop, followed by a two second delay.
- 4) Acceleration to the Programmed Speed opposite to the requested direction.
- 5) Run opposite the requested direction until the Home Input transitions from Active to Inactive.
- 6) Deceleration to the Starting Speed and stop, followed by a two second delay.
- 7) Return to the Home Input at the configured Starting Speed. Stop when the Home Input transitions from inactive to active.

Note ➤ If the overtravel limit is active when the Find Home Command is active, the profile will begin at step 4.

Controlling Home CW/CCW Commands In Progress

Controlled Stop Conditions

- The move completes without error.

Note ➤ The Hold_Move control bit in the Network Output Data cannot be used to bring a Home CW/CCW command to a controlled stop.

Immediate Stop Conditions

- The Immediate Stop bit makes a 0→1 transition in the Network Output Data.
- An inactive-to-active transition on an input configured as an E-Stop Input.
- The overtravel limit associated with travel in the opposite direction is activated. i.e. Activating the CCW limit during a CW Find Home command. This can occur if the overtravel limits are not wired to the IANF2(E) correctly, or if both overtravel limits are activated while the unit is trying to find the home position.
- The time defined by the Homing Timeout parameter in the configuration data is exceeded.

TASK 1: INSTALLING THE IANF2(E)

Introduction

The IANF2E can be installed as a single device. An IANF2 module must be installed as part of an AnyNET-I/O Stack, which includes an IANG1E or IANF2E as the network connection for the AnyNET-I/O Stack.

1.1 Safe Handling Guidelines

1.1.1 Prevent Electrostatic Damage

Caution Electrostatic discharge can damage the IANF2(E). Follow these guidelines when handling the module.

- 1) Touch a grounded object to discharge static potential before handling the module.
- 2) Work in a static-safe environment whenever possible.
- 3) Wear an approved wrist-strap grounding device when available.
- 4) Do not touch the pins of the bus connector or I/O connector.
- 5) Do not disassemble the module.
- 6) Store the unit in its anti-static bag and shipping box when it is not in use.

1.1.2 Prevent Debris From Entering the Unit

Warning While mounting devices, be sure that all debris (metal chips, wire strands, tapping liquids, etc.) is prevented from falling into the module. Debris may cause damage to the unit or unintended machine operation with possible personal injury.

1.1.3 Remove Power Before Servicing

Warning Remove power before removing or installing any IANF2(E) module. The module interconnect bus is not power limited.

1.2 Mounting

1.2.1 Dimensions

Figure T1.1 shows the dimensions of an IANF2(E) module.

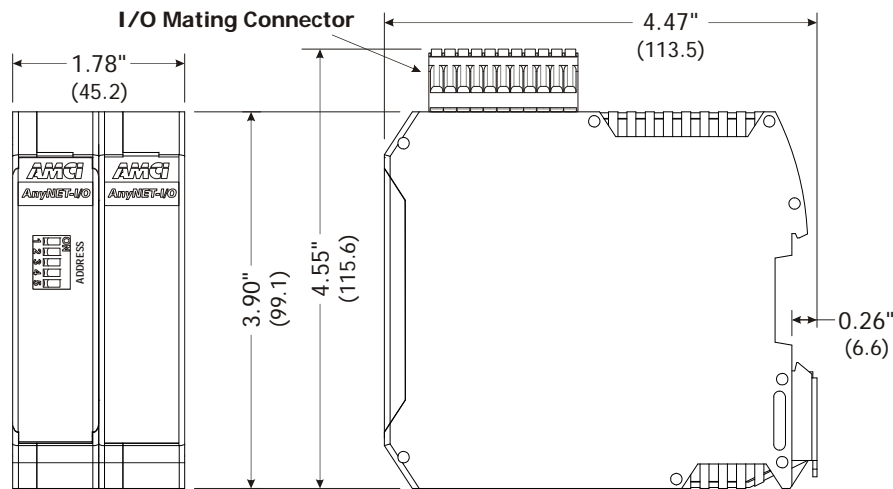


Figure T1.1 AnyNET-I/O Outline

1.2.2 Minimum Spacing

As shown in figure T1.2, you must maintain a minimum spacing of 2 inches (50 millimeters) from enclosure walls, wireways, adjacent equipment, etc. for adequate system ventilation.

Also note that the modules must be mounted in the orientation shown in the figure. Mounting the system in any other orientation will decrease the efficiency of the ventilation slots on the top and bottom of each module which may lead to system overheating and malfunction.

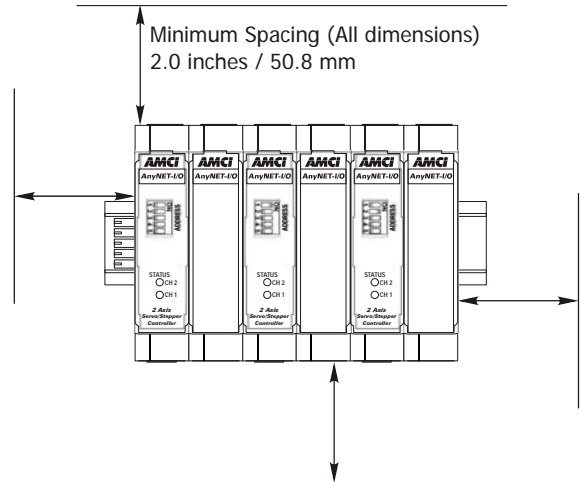


Figure T1.2 Ventilation Spacing

1.2.3 DIN Rail Installation

The IANF2(E) module is designed to be mounted on DIN rail. EN 05 022 - 35 x 7.5 DIN rail is preferred. EN 05 022 - 35 x 15 DIN rail can be used if mounting a single IANF2E module. The DIN rail for the modules should be securely installed and grounded before the modules are mounted on it.

1.2.4 Installing IIC-5 Connectors (as needed)

If you are installing a stack of modules instead of a single IANF2E, then you need to install the IIC-5 connectors on the DIN rail to allow the modules in the stack to communicate. Each module requires two IIC-5 connectors. Figure T1.3 shows how to install the IIC-5 connectors in the DIN rail.

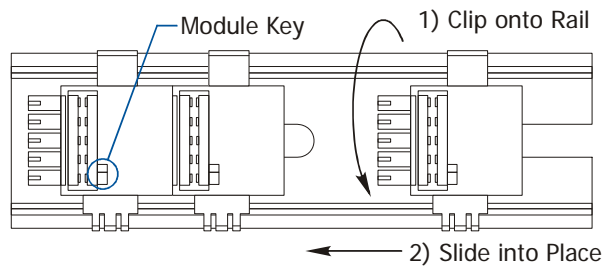


Figure T1.3 IIC-5 Connector Installation

Note▶ 1) EN 05 022 - 35 x 7.5 DIN rail must be used. The IIC-5 connectors are not properly supported in EN 05 022 - 35 x 15 DIN rail.
 2) Note the orientation of the IIC-5 connectors when installing them. The module key goes towards the bottom of the DIN rail.
 3) Two IIC-5 connectors are included with the IANF2(E). Additional connectors can be ordered from IDEC or directly from Phoenix Contact. Their Phoenix Contact part number is 271 37 22. (ME 22,5 TBUS 1,5/ 5-ST-3,81 KM)

1.2.5 Mounting the IANF2(E) Module

Mounting an AnyNET-I/O module is a very simple process thanks to the design of the enclosure.

- 1) Partially engage the IIC-5 connectors into the enclosure.
- 2) Engage the top clip in the enclosure with the top of the DIN rail and rotate the module down until the metal bracket snaps on to the DIN Rail.

Once all of your modules are installed, it is suggested to use the end caps from Phoenix Contact with the part number of 271 37 80 to secure the modules on the DIN Rail. These end caps prevent the module from sliding along the DIN rail if it is subjected to shock or vibration during machine operation.

1.3 Addressing

Each module needs to be given an address within the stack before the system will operate correctly. The address is set with the five position DIP switch on the front of the module. If you are using a single IANF2E module, its address must be set to zero, with all switches in their "OFF" position.

Note ►

- 1) Only a single switch should be in the "ON" position when setting the address.
- 2) The module that has an address of zero must have a network interface and it is the only module in the stack that can have a direct connection to the network.
- 3) If a module with a network interface has a non-zero address, then its network interface is disabled. This allows two or more IANF2E modules to work in a single stack.

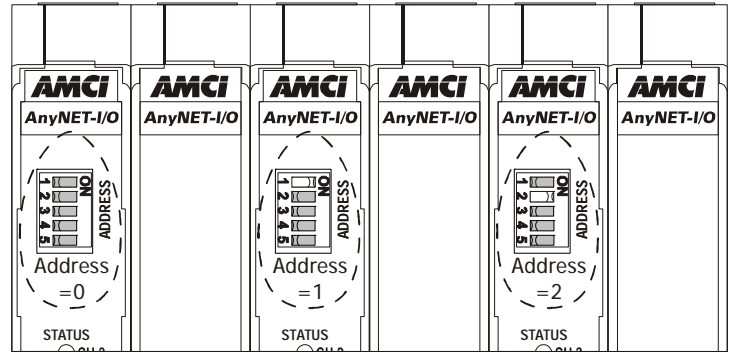


Figure T1.4 Addressing Example

Figure T1.4 is a close up of three modules in an AnyNET-I/O Stack. The module on the left is a module with a network interface, and has an address of zero (All DIP switches off.) This module has the active network interface and connects the stack to the network. Reading left to right, the remaining modules have addresses of one and two respectively. These modules *may* have network interfaces. If they do, their network interfaces are disabled.

1.4 Ethernet Connections

The Ethernet port is located on the bottom of the IANF2(E) module. The connector is a standard RJ-45 jack that will accept any standard or shielded 100baseT cable. (CAT5, CAT5e, CAT6, etc.) The Ethernet port on the module has an "auto switch" capability which eliminates the need for a crossover cable when directly connecting the module to a PC. A standard Ethernet cable can be used when connecting the IANF2(E) to any device.

IANF2E Bottom View

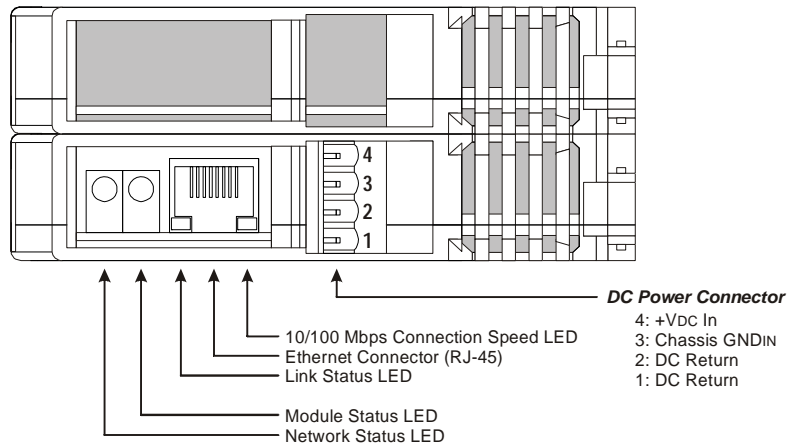


Figure T1.5 Ethernet Port Location

1.5 Power Wiring

1.5.1 Required Power

The IANF2(E) requires 12 to 24 Vdc \pm 20% supply to power the module. Surges to 30 Vdc will not damage it. Current draw is 120 mA @ 24 Vdc (2.9W).

This current draw does not include current through the inputs or outputs. If using the same supply to power the I/O, their load must be considered when sizing the supply.

- Inputs require 10 mA each.
- Differential outputs can supply up to 20 mAdc each.
- Single ended outputs can sink up to 30 mAdc each.

1.5.2 Wiring

As shown in the figure T1.5 on the previous page, the power connector is located on the bottom of the module. The mating connector is included with the IANF2(E). Spares are available from IDEC under the part number IMS-4M. They are also available from Phoenix Contact under their part number 187 80 37.

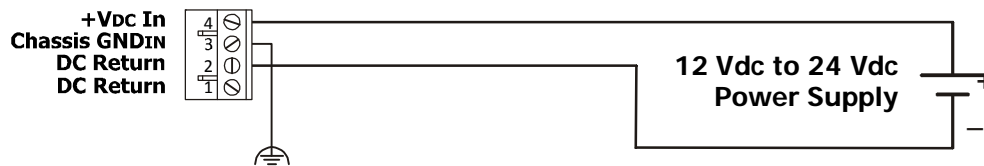


Figure T1.6 Power Connector Location

Power connections should be tight, as loose connections may lead to arcing which will heat the connector. Phoenix Contact specifies a tightening torque of 4.4 to 5.4 lb-in (0.5 to 0.6 Nm).

The power supply is connected to the pins marked "+Vdc In" and "DC Return". The "Chassis GND" pin is used to attach the IANF2(E) to earth ground.

Note ➤ 1) AnyNET-I/O modules are electrically isolated from the DIN rail by their mounting, but the Chassis GND connection is common to all of the modules in the stack through a pin of the IIC-5 connector. At least one module in the AnyNET-I/O Stack must be attached to earth ground through a heavy gauge stranded wire to ensure reliable operation of the stack.

2) Each IANF2(E) module must have its own power connection.

3) AnyNET-I/O modules will not be damaged if it is plugged into the InterConnect bus while power is applied, but the stack will not reconfigure itself to accept the new module.

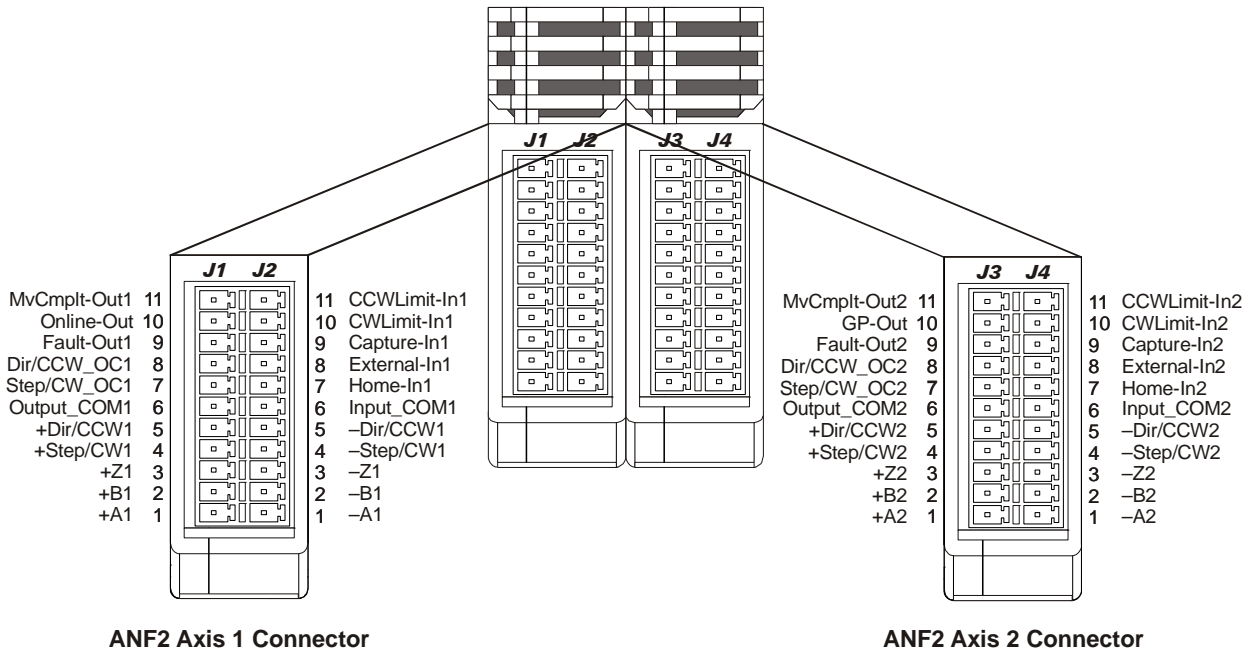


Caution

It is possible to use IANF2(E) and IANG1(E) modules in the same AnyNET-I/O Stack. The IANG1(E) modules are powered with 24 to 48 Vdc supplies. Applying 48Vdc to an IANF2(E), whether on the four pin power connector on the bottom of the module, or the twenty-two pin connector on the top, will damage the IANF2(E) module.

1.6 I/O Connector Pinout

Figure R5.4 below shows the pin out of the I/O connectors on the IANF2(E) modules. All mating connectors are included with the module. Additional connectors can be order for IDEC under the part number IMS-2x11.



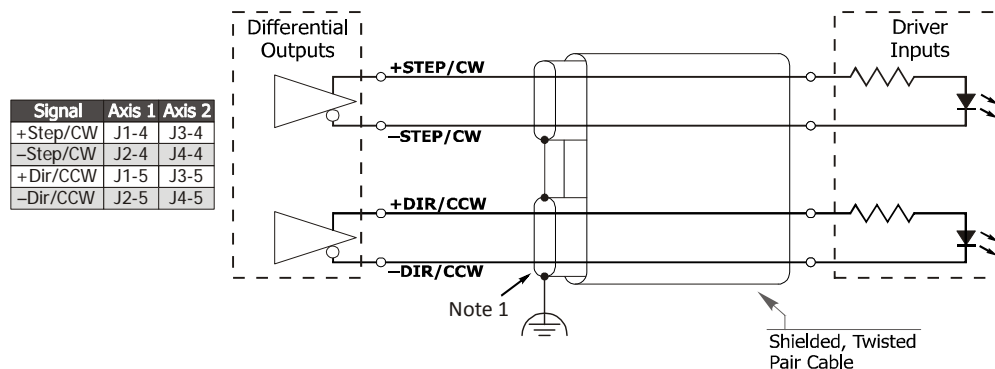
1.7 Drive Wiring

Connections to your stepper or servo drive can be made through the module's differential outputs or the single-ended open collector sinking outputs. You cannot use both output types at the same time.

1.7.1 Differential Wiring

Differential outputs are low voltage, low current outputs that require shielded, twisted pair cable to carry the signals. A suggested cable is the Belden 9729, which supports a 1 MHz pulse frequency to over 1000 feet of cable length; but any quality, shielded, twisted pair cable should suffice. A simple wiring diagram is shown below.

Differential Output From IANF2(E)



Note 1) Because they are low power signals, cabling should be made using a twisted pair cable with individual shields. The shield should be grounded to the same ground bus as the IANF2(E). The shield should not be grounded at both ends of the cable to avoid the potential of forming a ground loop.

1.7.2 Open Collector Wiring (Sourcing Inputs on Drive)

All singled ended outputs in the IANF2(E) are open collector sinking. They share a single common pin that must be tied to the ground connection of the power supply powering the outputs. Figure T1.7 below shows how to wire the outputs to open collector sourcing inputs on your drive.

IANF2(E) Single Ended Output Connection to Sourcing Load

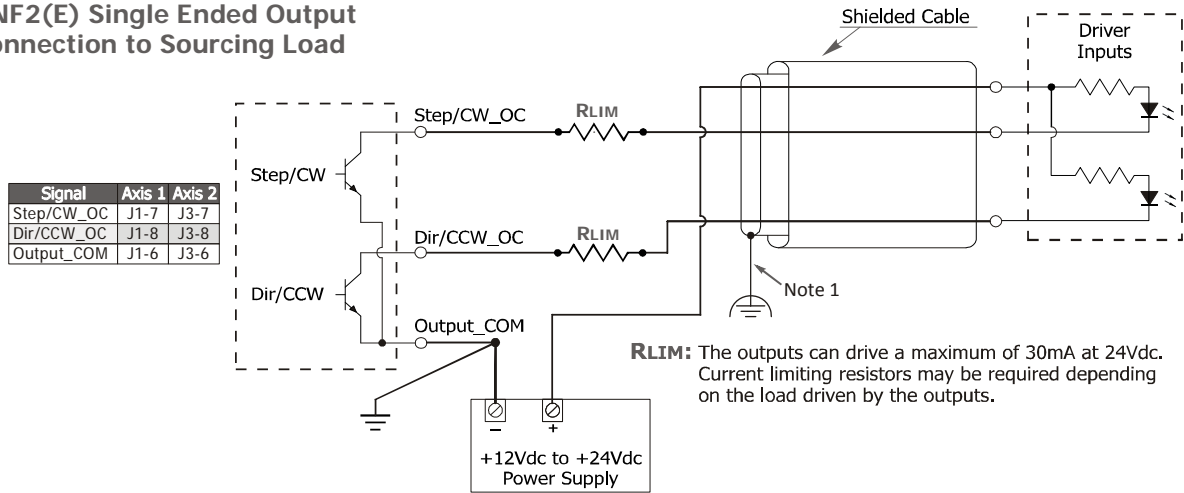


Figure T1.7 Open Collector Drive Wiring (Sourcing Inputs)

Note ▶ 1) Because they are low power signals, cabling should be made using a cable with an overall shield. The shield should be grounded to the same ground bus as the IANF2(E). The shield should not be grounded at both ends of the cable to avoid the potential of forming a ground loop.

2) RLIM may be needed to limit current flow. The IANF2(E) drivers can conduct a maximum of 30 milliamperes. The exact value of RLIM is based on the power supply voltage and the current requirements of the inputs on the drive.

1.7.3 Open Collector Wiring (Sinking Inputs on Drive)

All singled ended outputs in the IANF2(E) are open collector sinking. They share a single common pin that must be tied to the ground connection of the power supply powering the outputs. It is possible to wire the IANF2(E) outputs to open collector sinking inputs on a drive by using pull up resistors. This configuration will reverse the logic states of the outputs. An input on the drive will be normally on and turn off when the IANF2(E) output is in a logic "1" state. Figure T1.8 below shows how to wire the outputs to open collector sinking inputs on your drive.

IANF2(E) Single Ended Output Connection to Sinking Load

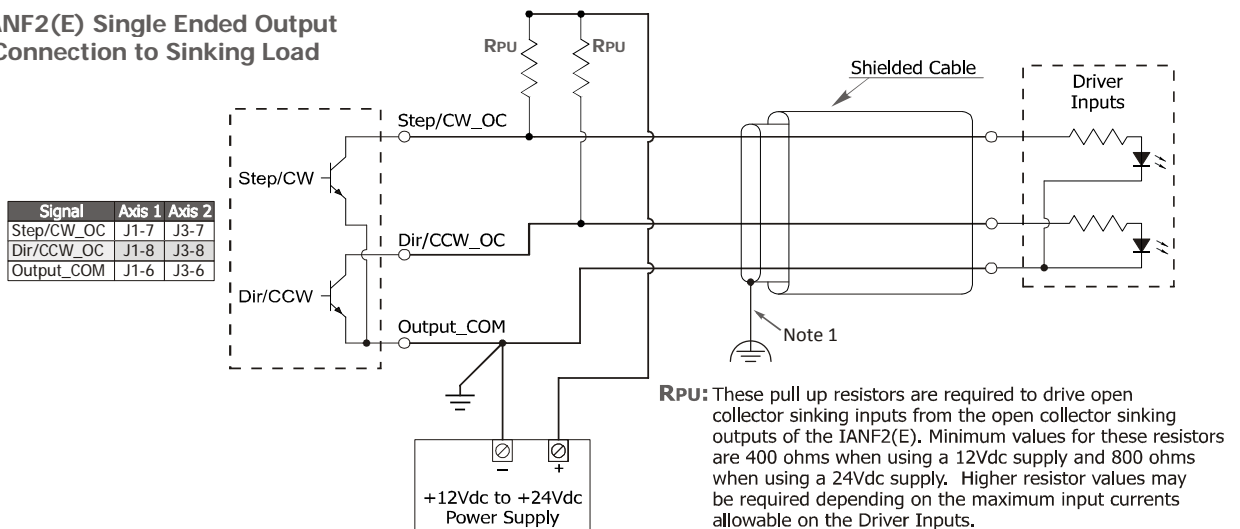


Figure T1.8 Open Collector Drive Wiring (Sinking Inputs)

Note 1) Because they are low power signals, cabling should be made using a cable with an overall shield. The shield should be grounded to the same ground bus as the IANF2(E). The shield should not be grounded at both ends of the cable to avoid the potential of forming a ground loop.



Caution

Carefully select the value of the pull up resistors. (Rpu in the figure.) The IANF2(E) outputs can conduct a maximum of 30 milliamperes without damage, so the minimum resistor value is 400 ohms / 0.5 watt when using a 12 Vdc supply and 800 ohms / 1.0 watt when using a 24 Vdc supply. You must increase the size of these resistors if the inputs to your drive cannot conduct 30 milliamps without damage.

1.8 Digital Output Wiring

Each axis has three digital outputs. Both axes have Move_Complete and Axis_Fault outputs. Axis 1 has an Online output, while axis 2 has a general purpose output. These outputs are electrically identical to the open collector Step/CW_OC and Dir/CCW_OC outputs available on each axis. These digital outputs share a common ground pin with the driver outputs. Wiring is identical to the diagrams given in figures T1.7 and T1.8.

Note The digital outputs can, but do not have to, use the power supply that powers the open collector driver outputs. If you use two different supplies, the low side of the supplies must be tied together at pin J1-6 for axis 1 and J3-6 for axis 2. For this reason, the two supplies should be mounted close together and share the same Earth Ground bus to avoid ground loop issues.

1.9 Discrete Input Wiring

Inputs to the IANF2(E) are open collector sinking. Therefore, the preferred sensor has a sourcing output. Sensors with sinking outputs can be wired to the IANF2(E) with the addition of a pull up resistor.

All inputs share a common pin that must be attached to the power supply common for proper operation. Input voltage for an "off" condition is 0 to 5 Vdc. Input voltage for an "on" condition is 8 to 24 Vdc. Designed for 12 and 24 Vdc operation, up to 27 Vdc can be applied to the input without damage. The maximum input current is 15 mA @ 24 Vdc.

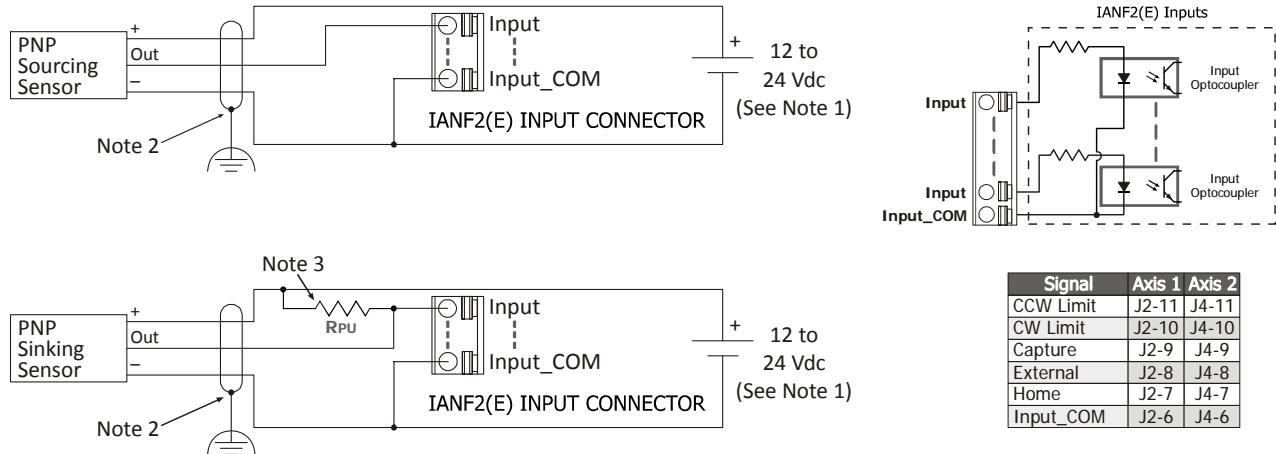


Figure T1.9 Discrete Input Wiring (Sourcing and Sinking Sensors)

Note 1) It is possible to use both 12 and 24 Vdc sensors, powered by two different supplies. In this configuration, the low side of the supplies must be tied together at pin J1-6 for axis 1 and J3-6 for axis 2. For this reason, the two supplies should be mounted close together and share the same Earth Ground bus to avoid ground loop issues.

2) Because they are low power signals, cabling from the sensor to the IANF2(E) should be done using a twisted pair cable with an overall shield. The shield should be grounded at the end when the signal is generated, which is the sensor end. If this is not practical, the shield should be grounded to the same ground bus as the IANF2(E).

3) See section 1.9.1 below.

1.9.1 Sinking Output Sensors Require a Pull Up Resistor

Sinking output sensors require an external pull up resistor because the input to the IANF2(E) also sinks current. Table T1.1 below shows the values of pull up resistors that will allow the IANF2(E) input to activate along with the current that the sensor must be able to sink when it is active.

Input Voltage	Pull Up Resistor	Sensor Current When Active
12 Vdc	1.4 kilohm	8.6 mA
24 Vdc	3.8 kilohm	6.3 mA

Table T1.1 Pull Up Resistor

The logical states of the sensor and IANF2(E) input will be reversed. The IANF2(E) input is off when the sensor is active. You can set the logic state of the IANF2(E) input when you configure the unit.

1.10 Encoder Wiring

1.10.1 Differential Encoder Wiring

The encoder inputs on the IANF2(E) accept 5Vdc differential signals from a quadrature incremental encoder. Figure T1.10 below shows a typical encoder installation.

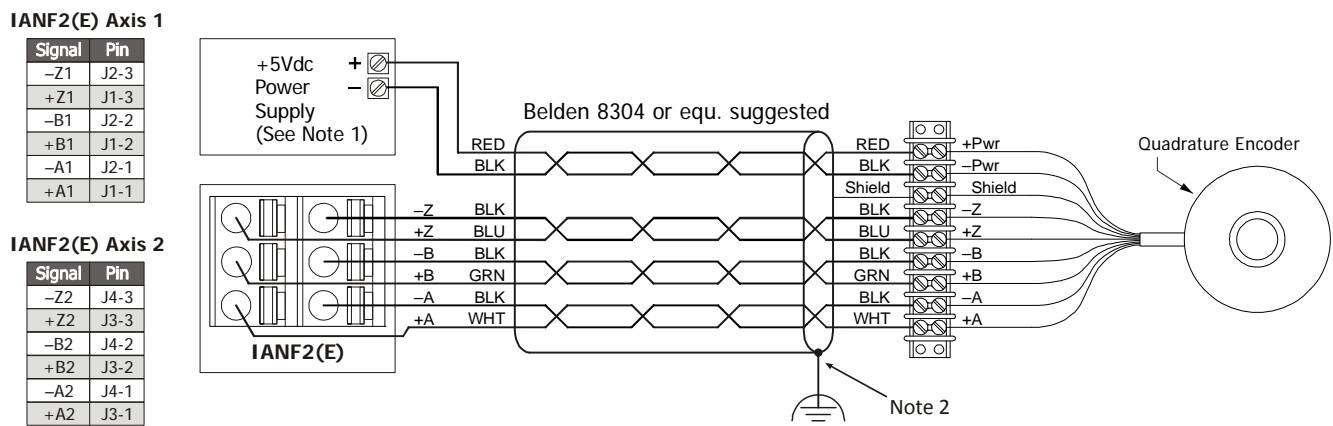


Figure T1.10 Differential Encoder Wiring

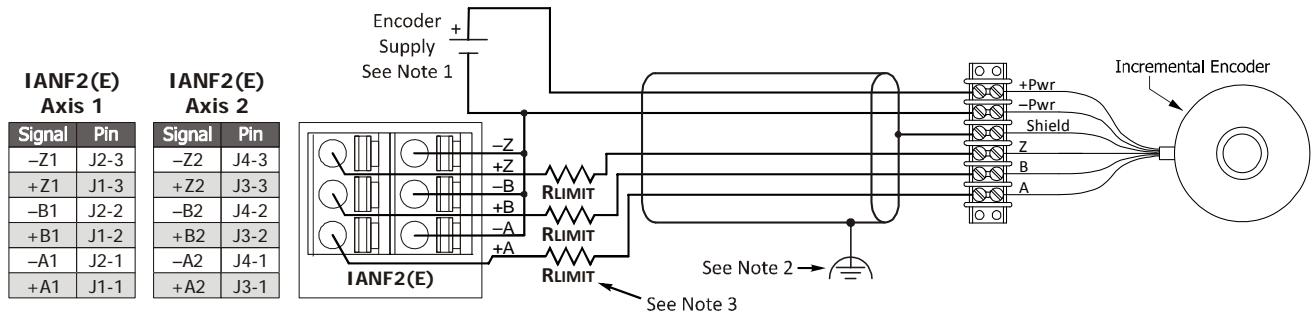
Note 1) The differential inputs on the IANF2(E) are designed for 5 Vdc differential signals. If using a power supply that is greater than 5 Vdc, verify that the encoder output 5 Vdc signals. If the output voltage exceeds 5 Vdc, current limiting resistors will be needed.

2) Because they are low power signals, cabling from the encoder to the IANF2(E) should be done using a twisted pair cable with an overall shield. The shield should be grounded at the end when the signal is generated, which is the encoder end. If this is not practical, the shield should be grounded to the same ground bus as the IANF2(E).

1.10.2 Single Ended Wiring

Figure T1.11 below shows how to wire the encoder inputs to both a single ended sourcing and single ended sinking encoder outputs.

IANF2(E) to Encoder with Sourcing Outputs



IANF2(E) to Encoder with Sinking Outputs

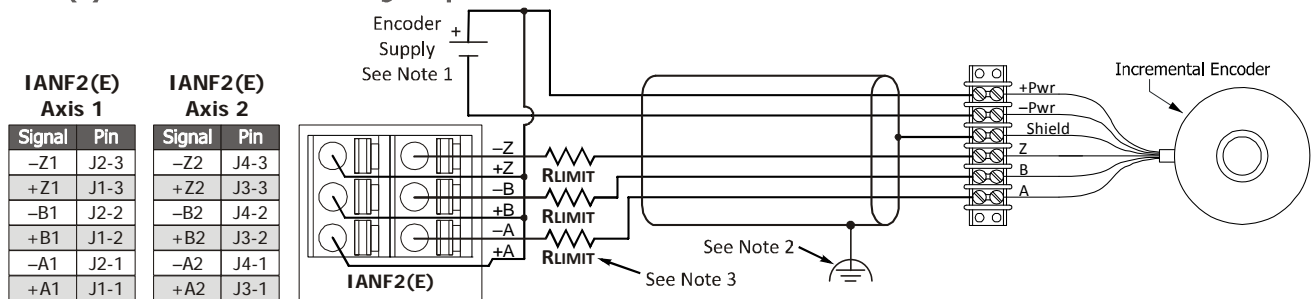


Figure T1.11 Single Ended Encoder Wiring

- Note** ▶
- 1) The encoder inputs on the IANF2(E) are designed for 5 Vdc signals. If using a power supply that is greater than 5 Vdc, verify that the encoder output 5 Vdc signals. If the output voltage exceeds 5 Vdc, current limiting resistors will be needed.
 - 2) Because they are low power signals, cabling from the encoder to the IANF2(E) should be done using a cable with an overall shield. The shield should be grounded at the end when the signal is generated, which is the encoder end. If this is not practical, the shield should be grounded to the same ground bus as the IANF2(E).
 - 3) See 1.10.2.1 below

1.10.2.1 Current Limiting Resistors

If the voltage on the outputs of a single ended encoder exceeds 5 Vdc, current limiting resistors must be added to the circuit to prevent damage to the encoder inputs on the IANF2(E). Table T1.2 below lists suggested values.

Input Voltage	Pull Up Resistor
5 Vdc	None
12 Vdc	1.0 kilohm
15 Vdc	1.2 kilohm
24 Vdc	2.7 kilohm

Table T1.2 current Limiting Resistor

1.11 Diagnostic Feedback Wiring

The IANF2(E) can be placed in a Diagnostic Feedback Mode that allows you to check the functionality of the differential motor control outputs and the encoder A and B inputs. You must wire the outputs to the inputs as shown in the table below before you can use this mode.

Axis #	Wire This Pin...		...To This Pin	
	Function	Pin #	Pin #	Function
1	+Step1	J1-4	J1-1	+A1
1	-Step1	J2-4	J2-1	-A1
1	+Direction1	J1-5	J1-2	+B1
1	-Direction1	J2-5	J2-2	-B1
2	+Step2	J3-4	J3-1	+A2
2	-Step2	J4-4	J4-1	-A2
2	+Direction2	J3-5	J3-2	+B2
2	-Direction2	J4-5	J4-2	-B2

Table R5.1 Diagnostic Feedback Wiring

The axis you are testing must be configured to output step and direction pulses. The IANF2(E) will issue a configuration error if the axis is configured to output CW/CCW pulses.

TASK 2: SET THE IP ADDRESS

Introduction

This section is intended for the engineer or technician responsible for setting the IP address of an IANF2E.

2.1 Determine the Best Method for Setting the IP Address

There are two methods for setting the IP address on an IANF2E. Table T2.1 below outlines the available methods and when you can use them.

Method	Restrictions	Starting page
<i>Use Factory Default Settings</i>	1) The machine must use 192.168.1.xxx subnet. 2) The 192.168.1.50 address must be available.	51
<i>Use the AMCI by IDEC Ethernet Tool</i>	No restrictions on use. The software can be used to set the IANF2E to any IPv4 address. The IP address will be stored in nonvolatile memory and used on subsequent power-ups.	51

Table T2.1 Methods for Setting the IP Address

Note ▶ There is a MAC address label on each IANF2E which has a writable surface. There is room on the label for writing the programmed IP address of the unit. It is a best practice to use this label to document the IP address of the unit in case it is ever repurposed.

2.2a Use Factory Default Settings

The factory default address for the IANF2E is 192.168.1.50 with a subnet mask of 255.255.255.0. The easiest way to verify this address is with the ping command as described in steps A.3 and A.4 of the Optional Task *Configure Your Network Interfaces*. The instructions for this optional task starts on page 109.

If the IANF2E does not respond to this address then it may take some effort to determine the correct address. There is a label on the drive that lists the MAC address of the device. There is space on the label for noting the IP address of the device if it is changed. If the address was not documented, a program called Wireshark (<https://www.wireshark.org/>) can be used to determine the address of the IANF2E.

Task Complete

The remainder of this chapter can be skipped, as it contains the method of setting the IP address of an IANF2E.

2.2b Use the AMCI by IDEC Ethernet Tool

PREREQUISITE: You must know the present IP address. The factory default address is 192.168.1.50.

PREREQUISITE: Task 1.4: *Ethernet Connections* found on page 43. You must be able to attach the IANF2E to your computer.

PREREQUISITE: Task 1.5: *Power Wiring* found on page 44. You must be able to power the IANF2E.

PREREQUISITE: Optional Task A: *Configure Your Network Interfaces*. (page 109) The network interfaces on your computer must be correctly configured before you can communicate with an IANF2E.

2.2b.1 If need be, download the AMCI by IDEC Ethernet Tool

- 1) Download link: http://us.idec.com/productdocuments/english/Other/AMCI-IEDEC-Configurator_1_2.zip
- 2) Unzip the file to any location on your computer.

2.2b.2 If need be, install the AMCI by IDEC Ethernet Tool

Once downloaded, simply extract the program from the ZIP file and run the program to install the AMCI by IDEC Ethernet software tool on your computer. The software installs as most products do, giving you the option to change the file locations before installing the utility. Once the install is complete, a link to the utility is available on the Start Menu. The install process only copies the utility to the designated location and creates links to the Start Menu. No changes are made to the registry settings of the computer.

2.2b.3 Verify that Your IDEC Controller is Disconnected from the IANF2E

Successfully changing the IP address of the IANF2E is fundamental to the operation of the unit. While performing this operation, the computer that is running the Ethernet tool should be the only device that can establish communications with the IANF2E.

2.2b.4 Apply or Cycle Power to the IANF2E

Cycling power to the IANF2E will reset any connections it may have had with the IDEC host controller.

2.2b.5 Start the AMCI by IDEC Ethernet Tool

Double click on the utility's icon. A welcome screen similar to the one in figure T2.1 below will appear.

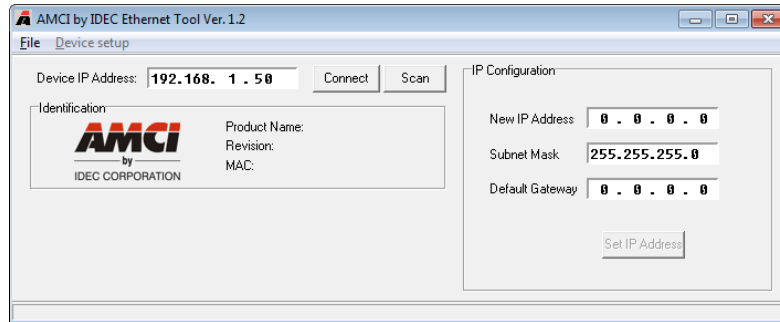


Figure T2.1 Ethernet Tool Welcome Screen

2.2b.6 Press the [SCAN] button and Connect to the IANF2E

Pressing the [Scan] button will open the window shown in figure T2.2. The IANF2E will appear in the scan list after a few seconds only if the unit and your network interface are on the same subnet.

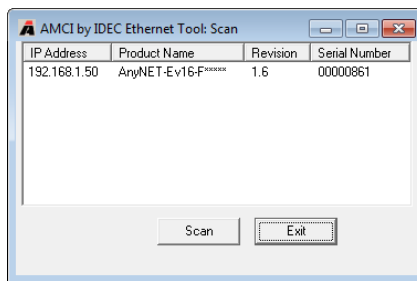


Figure T2.2 Scan for IANF2E

Once the IANF2E appears in the list, double click on the IP address. The Ethernet Tool will connect to the unit.

Optionally, from the main screen, you can enter the IP address of the IANF2E and press the [Connect] button. If the unit is found, the Ethernet Tool will directly connect with the unit.

2.2b.7 Set the IP Address, Subnet Mask, and Default Gateway

Enter your desired values into the IP Address, Subnet Mask, and Default Gateway fields.

Note ▶ The Default Gateway setting is not optional! It must be set to a valid address on the chosen subnet. If you do not have a required value for it, set the Default Gateway to the IP address of your FC6A controller.

2.2b.8 Write the New IP Address to the IANF2E

Click on the [Set IP Address] button. If there is an error in the settings, the utility will tell you what is wrong. Once all parameter values are correct, the utility will write the new IP address settings to the unit. These settings are saved to nonvolatile memory.

2.2b.9 Remove Power from the IANF2E

The new IP address will not be used until power to the IANF2E has been cycled.

Task Complete

TASK 3: WINDLDR 8.5 PROJECT SETUP

Introduction

The following task adds a module stack to a WindLDR 8.5+ project. (WindLDR versions 8.5+ is required.) A module stack consists of an IANF2E and zero to five additional IANF2 modules. This involves adding the IANF2E to the Remote Host List and then configure Modbus TCP connection requests.

Note Before adding an IANF2E to a new project, the controller must be defined. Refer to IDEC manuals and help files for information on accomplishing this task.

3.1 Add the IANF2E to the Remote Host List

3.1.1 Bring up the project window in the Work Space area on the left of the window.

If need be, select the View Tab, and verify that the Project Window icon is enabled. (Property Sheet and Cross Reference are disabled in figure T3.1 below)

Select the Project Window tab in the bottom of the Work Space area.

3.1.2 Double click on the Remote Host List in the Project Window to open the Remote Host List screen.

3.1.3 Click on the [New] button to open the Remote Host screen.

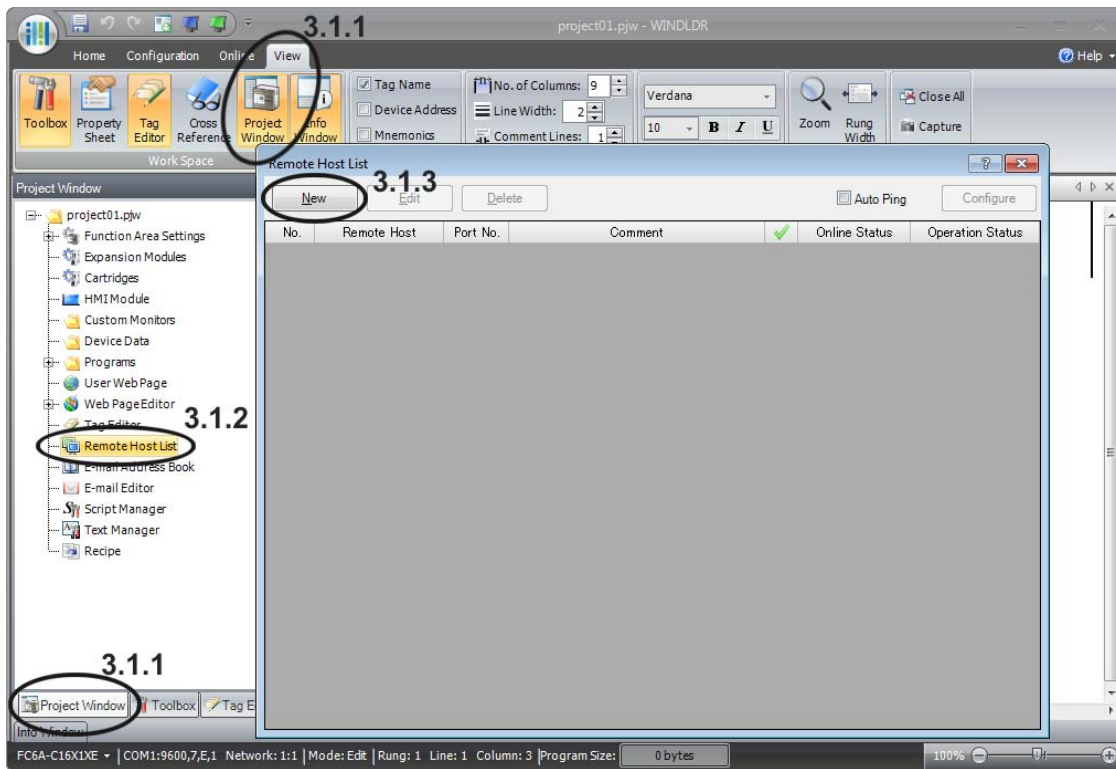


Figure T3.1 WindLDR Remote Host List

3.1.4 Enter the IP Address Information.

Enter the IP Address of the IANF2E. The default address is 192.168.1.50. You can enter a Host Name if you have configured a DNS server for the IDEC controller, which is beyond the scope of this manual.

Set the *Port*: field to 502, which is the default port used by the Modbus TCP protocol

You can add a *Comment*:, such as the name of the IANF2E stack in the project. This step is optional.

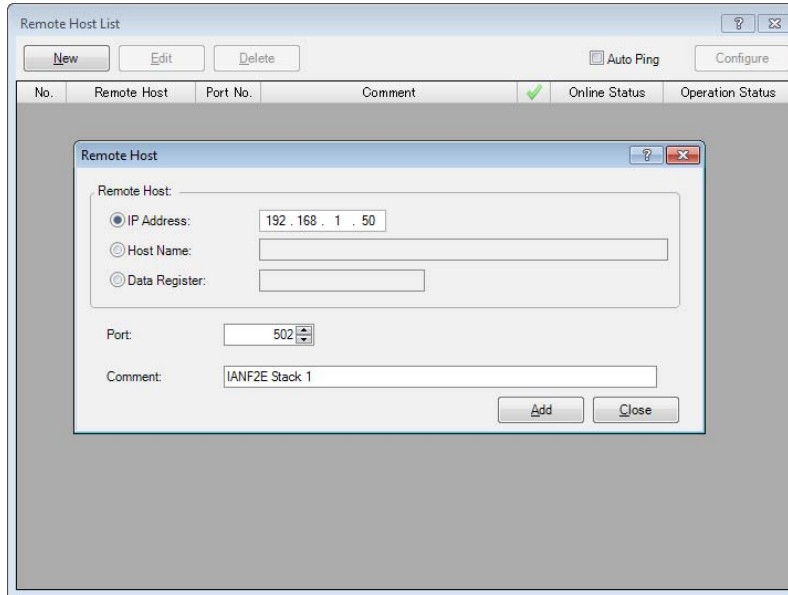


Figure T3.2 Remote Host List Entry Screen

3.1.5 Click on the *[Add]* button to add the IANF2E to the Remote Host List.

3.1.6 Repeat steps 3.1.3 and 3.1.4 for each IANF2E stack that must be added to the project. Each stack can consist of up to six modules.

3.1.7 Click on the *[Close]* button to close the Remote Host screen.

3.1.8 Close the Remote Host List screen.

3.2 Configure the Connection Settings for the IANF2E Stack

FC6A processors support a maximum of eight communication channels on the Ethernet port. At least one of the communications channels must be configured to service "Modbus TCP Client" communications.

3.2.1 From the WindLDR menu, select **Configuration > Connection Settings**.

The Function Area Settings dialog box will open, with the Connection Settings view selected. The eight communication channels are shown. Note that the default Communication Mode for all channels is "Maintenance Communication Server".

3.2.2 Set the Communications Mode for the selected channel to "Modbus TCP Client"

Click on the Communications Mode cell of the selected port. A drop down menu will appear. (See figure T3.3 below.) In this menu, select "Modbus TCP Client". The Modbus TCP Client dialog box will appear.

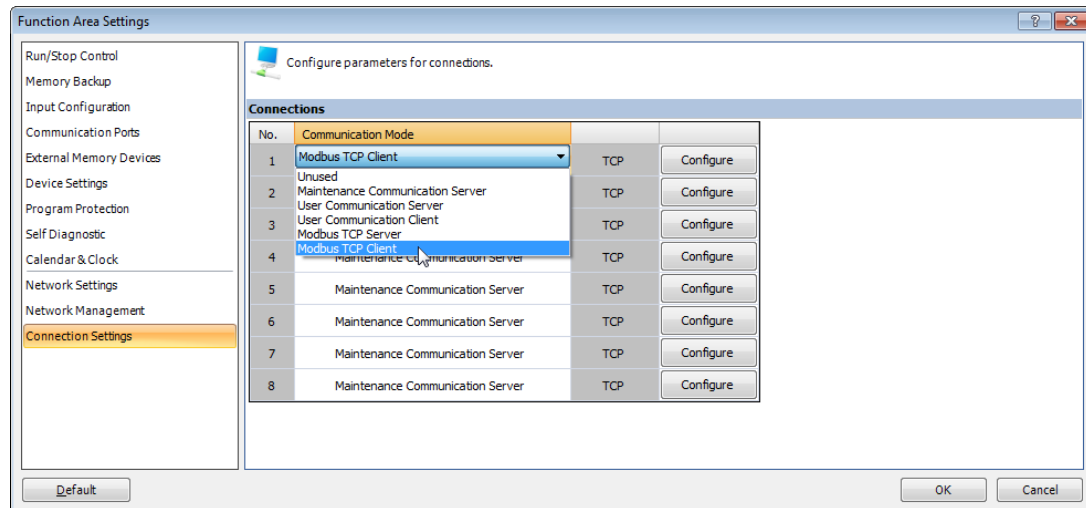


Figure T3.3 Connection Settings

3.2.3 Configure the Modbus TCP Client Requests

There are three setting areas that affect the communication channel as a whole.

Request Execution Settings: The "Request Execution Device" checkbox is typically left unchecked so that the FC6A communicates with the IANF2E stack continuously. If checked, communications is controlled by internal relays or data register bits. If your application requires you to enable this option, refer to the FC6A help file or the FC6A Series Communications Manual for additional information. The "Synchronize with auto ping" checkbox is typically left unchecked. If your application requires you to enable this option, refer to the FC6A help file or the FC6A Series Communications Manual for additional information.

Error Status: Error status information should be collected on each communication request. Select the "Use" radio button and set a data register address in the text box. All IDEC sample programs use register D49240 as the first error status register. The "Use a single DR for all communication requests" checkbox should be left unchecked so that error data on each request is collected individually. The number of data registers consumed will be equal to the number of requests on the channel, starting at the specified address. The "Update error status only when communication fails" checkbox is left unchecked in all IDEC sample programs. This setting allows the sample programs to act when communications is established.

Communication Settings: If needed, click on the [Communication Settings] button. A dialog box will open that allows you to set the Receive Timeout and Transmission Wait Time values. These settings are application specific and depend on network loading. Default values have been tested by AMCI and IDEC. Click on the [OK] button to close the dialog box.

In addition to these settings, two requests are required to communicate with each IANF2E stack. One request reads data from the stack and the other writes data to the stack. Each stack, which also can be considered to be each IP address, requires these two requests. For example, if you have two stacks on your machine, you will be configuring four requests.

Read Request: Set the following values to read data from the IANF2E stack.

Function Code: Click inside the cell and select "04 Read Input Registers".

Master Device Address: Depends on the first axis being controlled by the stack, which is the motor controlled by the module that connects the stack to the network. (IANF1E or IANF2E) See table T3.1 below to determine the correct starting address.

Machine Axis	Starting Data Register Address	Machine Axis	Starting Data Register Address
1	D49000	7	D49060
2	D49010	8	D49070
3	D49020	9	D49080
4	D49030	10	D49090
5	D49040	11	D49100
6	D49050	12	D49110

Table T3.1 Master Device Addresses: Read Input Registers

Data Size: Depends on the number of axes controlled by the AnyNET-I/O stack. Each IANF2(E) module controls a two axes. Each IANG1(E) module controls one axis. The Data Size must be set to (10 * number of axes in the AnyNET-I/O Stack.) For example, a stack of an IANF2E, a single IANF2 and two IANG1 modules controls a total of six axes. In this example the Data Size must be set to 60.

Remote Host No.: Click inside the cell to change it into a drop down menu control. Click on the down arrow and select the proper device that was configured in step 3.1, *Add the IANF2E to the Remote Host List*, above.

Slave Number: Not used by the AnyNET-I/O Stack. Set to "1".

Modbus Slave Address: 300001

The remaining fields are filled in automatically

Write Request: Set the following values to write values to the AnyNET-I/O Stack.

Function Code: Click inside the cell and select "16 Preset Multiple Registers".

Master Device Address: Depends on the first axis being controlled by the stack, which is the motor controlled by the module that connects the stack to the network. (IANG1E or IANF2E) See table T3.1 below to determine the correct address.

Machine Axis	Starting Data Register Address	Machine Axis	Starting Data Register Address
1	D49120	7	D49180
2	D49130	8	D49190
3	D49140	9	D49200
4	D49150	10	D49210
5	D49160	11	D49220
6	D49170	12	D49230

Table T3.2 Master Device Addresses: Read Input Registers

Data Size: Depends on the number of axes controlled by the AnyNET-I/O stack. Each IANF2(E) module controls a two axes. IANG1(E) modules control one axis. The Data Size must be set to (10 * number of axes in the AnyNET-I/O Stack.) For example, a stack of an IANF2E, a single IANF2 and two IANG1 modules controls a total of six axes. In this example the Data Size must be set to 60.

Remote Host No.: Click inside the cell to change it into a drop down menu control. Click on the down arrow to open the list and select the proper device that was configured in step 3.1, *Add the IANF2E to the Remote Host List*, above.

Slave Number: Not used by the AnyNET-I/O Stack. Set to "1".

Modbus Slave Address: 401025

The remaining fields are filled in automatically. Figure T3.4 below shows the Modbus TCP Client dialog box populated with a single IANF2E module as axis one and two of the machine.

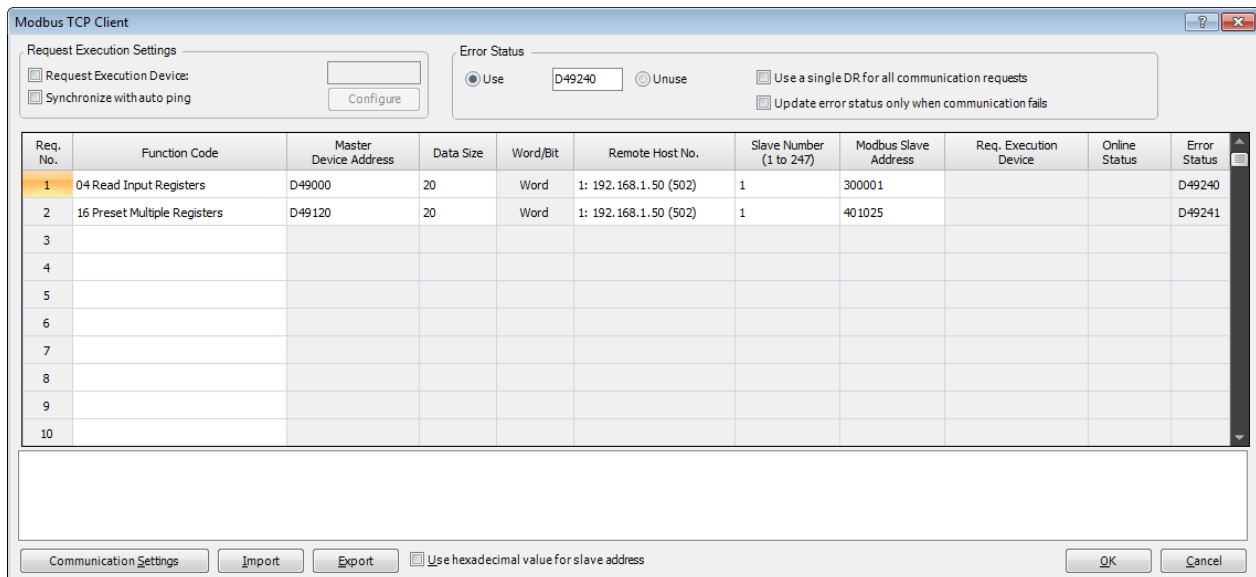


Figure T3.4 Modbus TCP Client Configuration Screen

TASK 4: INSTALLING CUSTOM MACROS

Introduction

IDEC has created User-defined Macros for configuring the IANF2(E) and most of its commands. These macros simplify the configuration and control of an IANF2(E). This task covers how to add these macros to a WindLDR 8.5+ project.

4.1 Download the Custom Macro File

- 1) Visit <http://us.idec.com/Home.aspx>
- 2) Click Products -> Automation -> AMCI Motion Controls
- 3) Select the *Controller* category
- 4) Click on the *Download* tab
- 5) Select the Download link for the *User Defined Macros - WindLDR* file.
- 6) Save the file to any location on your computer.

4.2 Add the Macros to a New or Existing Project

4.2.1 Open the Import User-defined Macro dialog box.

In the Project Window, if necessary, double click on Programs in order to make User-defined Macro visible. Right click on User-defined Macro and select Import from the resulting menu. In the Open dialog box, browse to the folder that contains the file you downloaded in step 4.1 above. Double click on the file name to open the Import User-defined Macro dialog box.

4.2.2 Import the macros into your program.

Click on the checkboxes next to the macros required for your project. You can import all of them if you so choose. Once the macros are selected, click on the [OK] button to import them.

Note ▶ You can repeat the steps in 4.2 to import additional macros at any time.

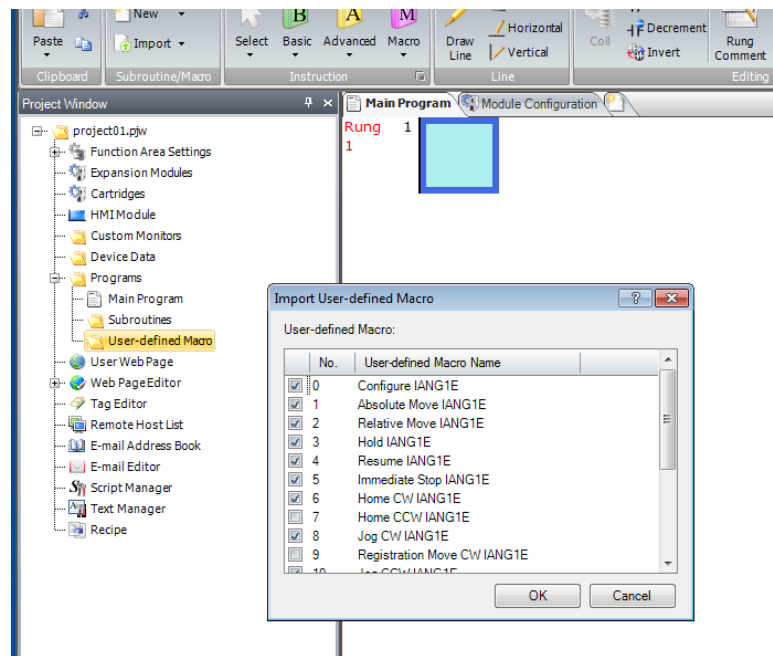


Figure T4.1 Importing User-defined Macros

Notes

TASK 5: USING A CUSTOM MACRO IN WINDLDR

Introduction

Once a User-defined Macro has been added to a project, it can be used as an instruction in a ladder logic program. This task covers how to add and configure a User-defined Macro in a ladder logic program.

5.1 Verify Memory Usage

The following address ranges are used by the IDEC supplied macros. You should check the registers currently being used in your project against this list and make adjustments as necessary. Note that registers D49000 through D49239 are the same addresses listed in the [WindLDR 8.5 Project Setup](#) Task starting on page 55.

Address Range			Use	Comments
D48000	to	D48999	Internal to Macros	These registers are used internally by the macros. Using registers in this range for any other purpose may result in unexpected operation.
D49000	to	D49009	Axis 1	Read Registers (Input data from the IANF2(E) modules. Updated through Modbus Function Code 04, Read Registers.) Data registers assigned to unused axes can be used for other purposes in the program.
D49010	to	D49019	Axis 2	
D49020	to	D49029	Axis 3	
D49030	to	D49039	Axis 4	
D49040	to	D49049	Axis 5	
D49050	to	D49059	Axis 6	
D49060	to	D49069	Axis 7	
D49070	to	D49079	Axis 8	
D49080	to	D49089	Axis 9	
D49090	to	D49099	Axis 10	
D49100	to	D49109	Axis 11	
D49110	to	D49119	Axis 12	
D49120	to	D49129	Axis 1	Preset Multiple Registers (Output data to the IANF2(E) modules. Updated through Modbus Function Code 16, Preset Multiple Registers.) Data registers assigned to unused axes can be used for other purposes in the program.
D49130	to	D49139	Axis 2	
D49140	to	D49149	Axis 3	
D49150	to	D49159	Axis 4	
D49160	to	D49169	Axis 5	
D49170	to	D49179	Axis 6	
D49180	to	D49189	Axis 7	
D49190	to	D49199	Axis 8	
D49200	to	D49209	Axis 9	
D49210	to	D49219	Axis 10	
D49220	to	D49229	Axis 11	
D49230	to	D49239	Axis 12	
D49240	to	D49499	Modbus TCP Communications	These registers used when monitoring and controlling Modbus TCP communications with the IANF2(E).
D49500	to	D49619	Read Registers Buffer	Memory space used to buffer the data read from the IANF2(E) modules. Any registers can be used, but all sample programs use addresses in this range. Note that this area is enough to buffer 12 axes. Unused registers can be used to store macro user data.
D49620	to	D49999	Macro User Data	Used to store data that is sent to an IANF2(E) through the macros. Any registers can be used, but all sample programs use addresses in this range.
M14500	to	M14747	Macro User Data	Addresses used to store data that is sent to an IANF2(E) through the macros. Any internal relays can be used, but all sample programs use internal relays in this range.
M14750	to	M14997	Program Control Bits	Internal relays used to store move states and control program flow. Any internal relays can be used, but all sample programs use internal relays in this range.

Table T5.1 Reserved Memory Addresses

5.1.1 Axis Addressing in an AnyNET-I/O Stack

The first axis in an AnyNET-I/O Stack is the first axis of the module that acts as the network connection for the stack. After that, the axes are numbered sequentially based on the number of axes in the module and the module's address within the stack. (This address is set with the DIP switches on the front of every module.) For example, assume an IANF2E followed by an IANG1 and an IANF2. The first two axes belong to the IANF2E, the third by the IANG1, and the fourth and fifth by the IANF2.

5.2 Verify/Alter Internal Relay Keep Settings

By default, the M internal relays are cleared at startup. The M14500 to M14747 internal relays listed above are used to store configuration and command bit settings and must be maintained between power cycles. Use the Configuration > Memory Backup screen to configure the FC6A to keep previous settings at startup.

5.2.1 Open the Function Area Settings Dialog Box.

From the WindLDR menu bar, select **Configuration > Memory Backup**. The **Function Area Settings** dialog box for Configure Keep/Clear Settings appears.

5.2.2 Verify or Alter the Keep Settings.

Under the 'M10000 to M17497' heading of the Internal Relay section, select either the 'Keep All' or 'Keep Specified Range' radio button. If your program does not rely on internal relays being reset on startup, then you can select 'Keep All'. If your program does rely on this behavior, select the 'Keep Specified Range' radio button and specify a range of M14500 to M14747.

Note ▶ You can only specify one range when using the 'Keep Specified Range' feature. If you are already using this feature, adjust this setting as necessary.

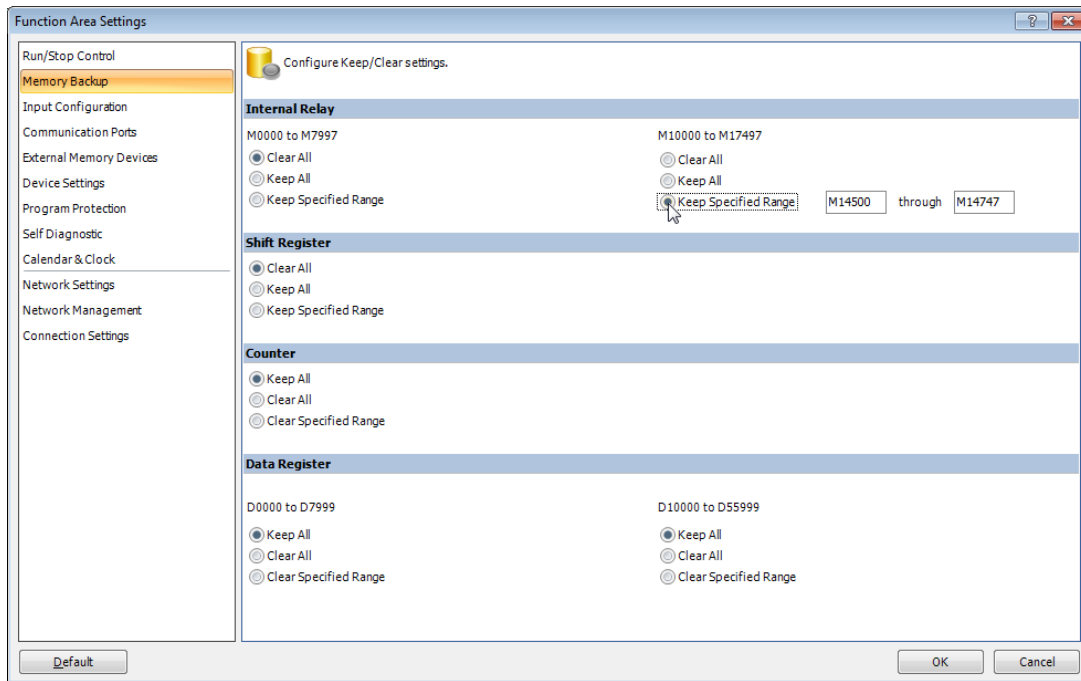


Figure T5.1 Internal Relay Keep Settings

5.3 Define the Memory Map to the Macro's Argument Devices

Macros contain a list of arguments that are used to pass data into, and out of, the macro. While adding the macro to your ladder logic, the list of arguments must be mapped to actual Device Addresses in the processor's memory. Sample programs available from IDEC use Device Addresses starting at register D49620 and internal relay M14500 for these purposes.

Even though it is possible to map from one register or relay to multiple macros, sample programs from IDEC use separate blocks of registers for each instance of a macro in the ladder logic program. This aids in debugging by preventing a change in value for one macro from inadvertently affecting another.

Before mapping registers and relays to their macros, tag name and comment fields can be added through the Tag Editor dialog box. This simplifies register lookup when adding them to the macro. The Tag Editor in the WindLDR program is used to add tag names and comments to the Device Addresses. Consult IDEC manuals and help files for information on using the Tag Editor.

5.3.1 Arguments to the "Configure IANF2" Macro

Argument	Device Type	Device Size	Range of Values	Description
A1	D (Data Register)	Word		Starting Address of the output registers assigned to the axis controlled by the IANF2(E). (D49120 to D49230 in the sample programs.) {Each IANF2(E) controls two consecutive axes in the data map, see table T5.1 on page 59}
A2	M (Internal Relay)	Bit	0/1	Capture_Input_Enable. 0 = Disabled, 1 = Enabled
A3	M (Internal Relay)	Bit	0/1	Capture_Input_Active_State. 0 = NC Input, 1 = NO Input
A4	M (Internal Relay)	Bit	0/1	External_Input_Enable. 0 = Disabled, 1 = Enabled
A5	M (Internal Relay)	Bit	0/1	Capture_Input_Active_State. 0 = NC Input, 1 = NO Input
A6	M (Internal Relay)	Bit	0/1	Home_Input_Enable. 0 = Disabled, 1 = Enabled
A7	M (Internal Relay)	Bit	0/1	Home_Input_Active_State. 0 = NC Input, 1 = NO Input
A8	D (Data Register)	Word	0, 1, 2	CW/Estop Input Select. 0 = No Used, 1 = CW Limit, 2 = E-Stop
A9	M (Internal Relay)	Bit	0/1	CW_Estop_Input_Active_State. 0 = NC Input, 1 = NO Input
A10	D (Data Register)	Word	0, 1, 2	CCW/Estop Input Select. 0 = No Used, 1 = CCW Limit, 2 = E-Stop
A11	M (Internal Relay)	Bit	0/1	CCW_Estop_Input_Active_State. 0 = NC Input, 1 = NO Input
A12	M (Internal Relay)	Bit	0/1	Backplane_Proximity_Bit_Enable 0 = Backplane Home Proximity Bit is not used 1 = Backplane Home Proximity Bit is used See Backplane Inputs found on page 37 for a description of this bit. This bit is reset in most applications.
A13	M (Internal Relay)	Bit	0/1	Quad_Encoder_Enable 0 = Quadrature Encoder Disabled, 1 = Quadrature Encoder Enabled
A14	M (Internal Relay)	Bit	0/1	Diagnostic_Feedback_Enable 0 = Diagnostic Feedback Disabled, 1 = Diagnostic Feedback Enabled
A15	M (Internal Relay)	Bit	0/1	Output_Pulse_Type 0 = CW/CCW Pulse Type, 1 = Step/DIR Pulse Type.
A16	M (Internal Relay)	Bit	0/1	Home_Operation_Type 0 = Home to Limit Switch (Set A6 and A7 appropriately for application.) 1 = Home to Z-Pulse (Set A13 = 1, set A12 appropriately for application.)
A17	D (Data Register)	Long†	1 to 1,000,000	Starting Speed. All moves begin and end at this speed
A18	D (Data Register)	Word	0 to 300	Homing Timeout. 0 to 300 seconds, 0 disables.

† Long integer values consume two consecutive data registers and the starting address must be even. For example, if argument A17 begins at address D49622, the next available address would be D49624. (A17 would consume both D49622 and D49623.) The starting register address of a long value must be even. (For example, argument A17 could not start with register D49621).

Table T5.2 Arguments to the "Configure IANF2E" Macro

5.3.2 Arguments to the "Absolute Move IANF2E" Macro

Argument	Device Type	Device Size	Range of Values	Description
A1	D (Data Register)	Word		Starting Address of the output registers assigned to the axis controlled by the IANF2(E). (D49120 to D49230 in the sample programs.) {Each IANF2(E) controls two consecutive axes in the data map, see table T5.1 on page 59}
A2	D (Data Register)	Long†	-1,073,741,823 to +1,073,741,823	Target Position
A3	D (Data Register)	Long†	Starting Speed to 1,000,000	Programmed Speed. (Starting Speed is one of the Arguments to the "Configure IANF2" Macro found on page 61.)
A4	D (Data Register)	Word	1 to 2000	Acceleration in steps/millisecond/second.
A5	D (Data Register)	Word	1 to 2000	Deceleration in steps/millisecond/second.
A6	D (Data Register)	Word	0, 1, or 2	Acceleration Type Select 0 = Linear Acceleration 1 = Triangular S-Curve Acceleration 2 = Trapezoidal S-Curve Acceleration

† Long integer values consume two consecutive data registers and the starting address must be even. For example, if argument A3 begins at address D49622, the next available address would be D49624. (A3 would consume both D49622 and D49623.) The starting register address of a long value must be even. (For example, argument A3 could not start with register D49621).

Table T5.3 Arguments to the "Absolute Move IANF2E" Macro

5.3.3 Arguments to the "Relative Move IANF2E" Macro

Argument	Device Type	Device Size	Range of Values	Description
A1	D (Data Register)	Word		Starting Address of the output registers assigned to the axis controlled by the IANF2(E). (D49120 to D49230 in the sample programs.) {Each IANF2(E) controls two consecutive axes in the data map, see table T5.1 on page 59}
A2	D (Data Register)	Long†	-1,073,741,823 to +1,073,741,823	Target Distance
A3	D (Data Register)	Long†	Starting Speed to 1,000,000	Programmed Speed. (Starting Speed is one of the Arguments to the "Configure IANF2" Macro found on page 61.)
A4	D (Data Register)	Word	1 to 2000	Acceleration in steps/millisecond/second.
A5	D (Data Register)	Word	1 to 2000	Deceleration in steps/millisecond/second.
A6	D (Data Register)	Word	0, 1, or 2	Acceleration Type Select 0 = Linear Acceleration 1 = Triangular S-Curve Acceleration 2 = Trapezoidal S-Curve Acceleration

† Long integer values consume two consecutive data registers and the starting address must be even. For example, if argument A3 begins at address D49622, the next available address would be D49624. (A3 would consume both D49622 and D49623.) The starting register address of a long value must be even. (For example, argument A3 could not start with register D49621).

Table T5.4 Arguments to the "Relative Move IANF2E" Macro

5.3.4 Arguments to the "Hold IANF2E" Macro

Argument	Device Type	Device Size	Range of Values	Description
A1	D (Data Register)	Word		Starting Address of the output registers assigned to the axis controlled by the IANF2(E). (D49120 to D49230 in the sample programs.) {Each IANF2(E) controls two consecutive axes in the data map, see table T5.1 on page 59}

Table T5.5 Arguments to the "Hold IANF2E" Macro

5.3.5 Arguments to the "Resume IANF2E" Macro

Argument	Device Type	Device Size	Range of Values	Description
A1	D (Data Register)	Word		Starting Address of the output registers assigned to the axis controlled by the IANF2(E). (D49120 to D49230 in the sample programs.) {Each IANF2(E) controls two consecutive axes in the data map, see table T5.1 on page 59}

Table T5.6 Arguments to the "Resume IANF2E" Macro

5.3.6 Arguments to the "Immediate Stop IANF2E" Macro

Argument	Device Type	Device Size	Range of Values	Description
A1	D (Data Register)	Word		Starting Address of the output registers assigned to the axis controlled by the IANF2(E). (D49120 to D49230 in the sample programs.) {Each IANF2(E) controls two consecutive axes in the data map, see table T5.1 on page 59}

Table T5.7 Arguments to the "Immediate Stop IANF2E" Macro

5.3.7 Arguments to the "Home CW IANF2E" Macro

Argument	Device Type	Device Size	Range of Values	Description
A1	D (Data Register)	Word		Starting Address of the output registers assigned to the axis controlled by the IANF2(E). (D49120 to D49230 in the sample programs.) {Each IANF2(E) controls two consecutive axes in the data map, see table T5.1 on page 59}
A2	D (Data Register)	Long†	Starting Speed to 1,000,000	Programmed Speed. (Starting Speed is one of the Arguments to the "Configure IANF2" Macro found on page 61.)
A3	D (Data Register)	Word	1 to 2000	Acceleration in steps/millisecond/second.
A4	D (Data Register)	Word	1 to 2000	Deceleration in steps/millisecond/second.

† Long integer values consume two consecutive data registers and the starting address must be even. For example, if argument A3 begins at address D49622, the next available address would be D49624. (A3 would consume both D49622 and D49623.) The starting register address of a long value must be even. (For example, argument A3 could not start with register D49621).

Table T5.8 Arguments to the "Home CW IANF2E" Macro

5.3.8 Arguments to the "Home CCW IANF2E" Macro

Argument	Device Type	Device Size	Range of Values	Description
A1	D (Data Register)	Word		Starting Address of the output registers assigned to the axis controlled by the IANF2(E). (D49120 to D49230 in the sample programs.) {Each IANF2(E) controls two consecutive axes in the data map, see table T5.1 on page 59}
A2	D (Data Register)	Long†	Starting Speed to 1,000,000	Programmed Speed. (Starting Speed is one of the Arguments to the "Configure IANF2" Macro found on page 61.)
A3	D (Data Register)	Word	1 to 2000	Acceleration in steps/millisecond/second.
A4	D (Data Register)	Word	1 to 2000	Deceleration in steps/millisecond/second.

† Long integer values consume two consecutive data registers and the starting address must be even. For example, if argument A3 begins at address D49622, the next available address would be D49624. (A3 would consume both D49622 and D49623.) The starting register address of a long value must be even. (For example, argument A3 could not start with register D49621).

Table T5.9 Arguments to the "Home CCW IANF2E" Macro

5.3.9 Arguments to the "Jog CW IANF2E" Macro

Argument	Device Type	Device Size	Range of Values	Description
A1	D (Data Register)	Word		Starting Address of the output registers assigned to the axis controlled by the IANF2(E). (D49120 to D49230 in the sample programs.) {Each IANF2(E) controls two consecutive axes in the data map, see table T5.1 on page 59}
A2	D (Data Register)	Long†	1 to 1,000,000	Programmed Speed.
A3	D (Data Register)	Word	1 to 2000	Acceleration in steps/millisecond/second.
A4	D (Data Register)	Word	1 to 2000	Deceleration in steps/millisecond/second.
A5	D (Data Register)	Word	0, 1, or 2	Acceleration Type Select 0 = Linear Acceleration 1 = Triangular S-Curve Acceleration 2 = Trapezoidal S-Curve Acceleration

† Long integer values consume two consecutive data registers and the starting address must be even. For example, if argument A3 begins at address D49622, the next available address would be D49624. (A3 would consume both D49622 and D49623.) The starting register address of a long value must be even. (For example, argument A3 could not start with register D49621).

Table T5.10 Arguments to the "Jog CW IANF2E" Macro

5.3.10 Arguments to the "Registration Move CW IANF2E" Macro

Argument	Device Type	Device Size	Range of Values	Description
A1	D (Data Register)	Word		Starting Address of the output registers assigned to the axis controlled by the IANF2(E). (D49120 to D49230 in the sample programs.) {Each IANF2(E) controls two consecutive axes in the data map, see table T5.1 on page 59}
A2	D (Data Register)	Long†	1 to 1,073,741,823	Stopping Distance.
A3	D (Data Register)	Long†	1 to 1,000,000	Programmed Speed.
A4	D (Data Register)	Word	1 to 2000	Acceleration in steps/millisecond/second.
A5	D (Data Register)	Word	1 to 2000	Deceleration in steps/millisecond/second.
A6	D (Data Register)	Word	0, 1, or 2	Acceleration Type Select 0 = Linear Acceleration 1 = Triangular S-Curve Acceleration 2 = Trapezoidal S-Curve Acceleration

† Long integer values consume two consecutive data registers and the starting address must be even. For example, if argument A3 begins at address D49622, the next available address would be D49624. (A3 would consume both D49622 and D49623.) The starting register address of a long value must be even. (For example, argument A3 could not start with register D49621).

Table T5.11 Arguments to the "Registration Move CW IANF2E" Macro

5.3.11 Arguments to the "Jog CCW IANF2E" Macro

Argument	Device Type	Device Size	Range of Values	Description
A1	D (Data Register)	Word		Starting Address of the output registers assigned to the axis controlled by the IANF2(E). (D49120 to D49230 in the sample programs.) {Each IANF2(E) controls two consecutive axes in the data map, see table T5.1 on page 59}
A2	D (Data Register)	Long†	1 to 1,000,000	Programmed Speed.
A3	D (Data Register)	Word	1 to 2000	Acceleration in steps/millisecond/second.
A4	D (Data Register)	Word	1 to 2000	Deceleration in steps/millisecond/second.
A5	D (Data Register)	Word	0, 1, or 2	Acceleration Type Select 0 = Linear Acceleration 1 = Triangular S-Curve Acceleration 2 = Trapezoidal S-Curve Acceleration

† Long integer values consume two consecutive data registers and the starting address must be even. For example, if argument A2 begins at address D49622, the next available address would be D49624. (A2 would consume both D49622 and D49623.) The starting register address of a long value must be even. (For example, argument A2 could not start with register D49621).

Table T5.12 Arguments to the "Jog CCW IANF2E" Macro

5.3.12 Arguments to the "Registration Move CCW IANF2E" Macro

Argument	Device Type	Device Size	Range of Values	Description
A1	D (Data Register)	Word		Starting Address of the output registers assigned to the axis controlled by the IANF2(E). (D49120 to D49230 in the sample programs.) {Each IANF2(E) controls two consecutive axes in the data map, see table T5.1 on page 59}
A2	D (Data Register)	Long†	1 to 1,073,741,823	Stopping Distance.
A3	D (Data Register)	Long†	1 to 1,000,000	Programmed Speed.
A4	D (Data Register)	Word	1 to 2000	Acceleration in steps/millisecond/second.
A5	D (Data Register)	Word	1 to 2000	Deceleration in steps/millisecond/second.
A6	D (Data Register)	Word	0, 1, or 2	Acceleration Type Select 0 = Linear Acceleration 1 = Triangular S-Curve Acceleration 2 = Trapezoidal S-Curve Acceleration

† Long integer values consume two consecutive data registers and the starting address must be even. For example, if argument A3 begins at address D49622, the next available address would be D49624. (A3 would consume both D49622 and D49623.) The starting register address of a long value must be even. (For example, argument A3 could not start with register D49621).

Table T5.13 Arguments to the "Registration Move CCW IANF2E" Macro

5.3.13 Arguments to the "Preset Position IANF2E" Macro

Argument	Device Type	Device Size	Range of Values	Description
A1	D (Data Register)	Word		Starting Address of the output registers assigned to the axis controlled by the IANF2(E). (D49120 to D49230 in the sample programs.) {Each IANF2(E) controls two consecutive axes in the data map, see table T5.1 on page 59}
A2	D (Data Register)	Long†	-1,073,741,823 to +1,073,741,823	Desired Motor Position value.

† Long integer values consume two consecutive data registers and the starting address must be even. For example, if argument A2 begins at address D49622, the next available address would be D49624. (A2 would consume both D49622 and D49623.) The starting register address of a long value must be even. (For example, argument A2 could not start with register D49621).

Table T5.14 Arguments to the "Preset Position IANF2E" Macro

5.3.14 Arguments to the "Clear Errors IANF2E" Macro

Argument	Device Type	Device Size	Range of Values	Description
A1	D (Data Register)	Word		Starting Address of the output registers assigned to the axis controlled by the IANF2(E). (D49120 to D49230 in the sample programs.) {Each IANF2(E) controls two consecutive axes in the data map, see table T5.1 on page 59}

Table T5.15 Arguments to the "Clear Errors IANF2E" Macro

5.3.15 Arguments to the "Preset Encoder Position IANF2E" Macro

Argument	Device Type	Device Size	Range of Values	Description
A1	D (Data Register)	Word		Starting Address of the output registers assigned to the axis controlled by the IANF2(E). (D49120 to D49230 in the sample programs.) {Each IANF2(E) controls two consecutive axes in the data map, see table T5.1 on page 59}
A2	D (Data Register)	Long†	-1,073,741,823 to +1,073,741,823	Desired Encoder Position value.

† Long integer values consume two consecutive data registers and the starting address must be even. For example, if argument A2 begins at address D49622, the next available address would be D49624. (A2 would consume both D49622 and D49623.) The starting register address of a long value must be even. (For example, argument A2 could not start with register D49621).

Table T5.16 Arguments to the "Preset Encoder Position IANF2E" Macro

5.3.16 Arguments to the "Encoder Follower CW IANF2E" Macro

Argument	Device Type	Device Size	Range of Values	Description
A1	D (Data Register)	Word		Starting Address of the output registers assigned to the axis controlled by the IANF2(E). (D49120 to D49230 in the sample programs.) {Each IANF2(E) controls two consecutive axes in the data map, see table T5.1 on page 59}
A2	D (Data Register)	Word	1 to 255	Encoder Follower Multiplier
A3	D (Data Register)	Word	1 to 255	Encoder Follower Divisor
A4	D (Data Register)	Word	1 to 2000	Acceleration in steps/millisecond/second.
A5	D (Data Register)	Word	1 to 2000	Deceleration in steps/millisecond/second.

Table T5.17 Arguments to the "Encoder Follower CW IANF2E" Macro

5.3.17 Arguments to the "Encoder Follower CCW IANF2E" Macro

Argument	Device Type	Device Size	Range of Values	Description
A1	D (Data Register)	Word		Starting Address of the output registers assigned to the axis controlled by the IANF2(E). (D49120 to D49230 in the sample programs.) {Each IANF2(E) controls two consecutive axes in the data map, see table T5.1 on page 59}
A2	D (Data Register)	Word	1 to 255	Encoder Follower Multiplier
A3	D (Data Register)	Word	1 to 255	Encoder Follower Divisor
A4	D (Data Register)	Word	1 to 2000	Acceleration in steps/millisecond/second.
A5	D (Data Register)	Word	1 to 2000	Deceleration in steps/millisecond/second.

Table T5.18 Arguments to the "Encoder Follower CCW IANF2E" Macro

5.3.18 Arguments to the "Set Min Reg Move Distance IANF2E" Macro

Argument	Device Type	Device Size	Range of Values	Description
A1	D (Data Register)	Word		Starting Address of the output registers assigned to the axis controlled by the IANF2(E). (D49120 to D49230 in the sample programs.) {Each IANF2(E) controls two consecutive axes in the data map, see table T5.1 on page 59}
A2	D (Data Register)	Long†	0 to +1,073,741,823	Minimum Registration Move Distance

† Long integer values consume two consecutive data registers and the starting address must be even. For example, if argument A2 begins at address D49622, the next available address would be D49624. (A2 would consume both D49622 and D49623.) The starting register address of a long value must be even. (For example, argument A2 could not start with register D49621).

Table T5.19 Arguments to the "Set Min Reg Move Distance IANF2E" Macro

5.3.19 Arguments to the "Linear Interpolated Move IANF2E" Macro

Note ▶ Interpolated moves control both axes of the IANF2(E) simultaneously. The starting address of the command block, (argument A1), is the starting address of the first axis controlled by the IANF2(E). The command block is twenty words in length.

Argument	Device Type	Device Size	Range of Values	Description
A1	D (Data Register)	Word		Starting Address of the output registers assigned to the first axis controlled by the IANF2(E). (D49120 to D49220 in the sample programs.) {Each IANF2(E) controls two consecutive axes in the data map, see table T5.1 on page 59}
A2	M (Internal Relay)	Bit	0 or 1	Move Type: 0 = Relative, 1 = Absolute
A3	D (Data Register)	Word	0, 1, or 2	Acceleration Type Select 0 = Linear Acceleration 1 = Triangular S-Curve Acceleration 2 = Trapezoidal S-Curve Acceleration
A4	D (Data Register)	Word	1 to 1,000,000	Programmed Interpolated Speed
A5	D (Data Register)	Word	1 to 2000	Interpolated Acceleration in steps/millisecond/second.
A6	D (Data Register)	Word	1 to 2000	Interpolated Deceleration in steps/millisecond/second.
A7	D (Data Register)	Long†	-1,073,741,823 to +1,073,741,823	Motor Position End Point X value. (First Axis)
A8	D (Data Register)	Long†	-1,073,741,823 to +1,073,741,823	Motor Position End Point Y value. (Second Axis)

† Long integer values consume two consecutive data registers and the starting address must be even. For example, if argument A2 begins at address D49622, the next available address would be D49624. (A2 would consume both D49622 and D49623.) The starting register address of a long value must be even. (For example, argument A2 could not start with register D49621).

Table T5.20 Arguments to the "Linear Interpolated Move IANF2E" Macro

5.3.20 Arguments to the "Via Point Circular Move IANF2E" Macro

Note▶ Interpolated moves control both axes of the IANF2(E) simultaneously. The starting address of the command block, (argument A1), is the starting address of the first axis controlled by the IANF2(E). The command block is twenty words in length.

Argument	Device Type	Device Size	Range of Values	Description
A1	D (Data Register)	Word		Starting Address of the output registers assigned to the first axis controlled by the IANF2(E). (D49120 to D49220 in the sample programs.) {Each IANF2(E) controls two consecutive axes in the data map, see table T5.1 on page 59}
A2	M (Internal Relay)	Bit	0 or 1	Move Type: 0 = Relative, 1 = Absolute
A3	D (Data Register)	Word	0, 1, or 2	Acceleration Type Select 0 = Linear Acceleration 1 = Triangular S-Curve Acceleration 2 = Trapezoidal S-Curve Acceleration
A4	D (Data Register)	Word	1 to 1,000,000	Programmed Interpolated Speed
A5	D (Data Register)	Word	1 to 2000	Interpolated Acceleration in steps/millisecond/second.
A6	D (Data Register)	Word	1 to 2000	Interpolated Deceleration in steps/millisecond/second.
A7	D (Data Register)	Long†	-1,073,741,823 to +1,073,741,823	Motor Position End Point X value. (First Axis)
A8	D (Data Register)	Long†	-1,073,741,823 to +1,073,741,823	Motor Position End Point Y value. (Second Axis)
A9	D (Data Register)	Long†	-1,073,741,823 to +1,073,741,823	Move's Via Point X value. (First Axis)
A10	D (Data Register)	Long†	-1,073,741,823 to +1,073,741,823	Move's Via Point Y value. (Second Axis)

† Long integer values consume two consecutive data registers and the starting address must be even. For example, if argument A2 begins at address D49622, the next available address would be D49624. (A2 would consume both D49622 and D49623.) The starting register address of a long value must be even. (For example, argument A2 could not start with register D49621).

Table T5.21 Arguments to the "Via Point Circular Move IANF2E" Macro

5.3.21 Arguments to the "Center Point Circular Move IANF2E" Macro

Note▶ Interpolated moves control both axes of the IANF2(E) simultaneously. The starting address of the command block, (argument A1), is the starting address of the first axis controlled by the IANF2(E). The command block is twenty words in length.

Argument	Device Type	Device Size	Range of Values	Description
A1	D (Data Register)	Word		Starting Address of the output registers assigned to the first axis controlled by the IANF2(E). (D49120 to D49220 in the sample programs.) {Each IANF2(E) controls two consecutive axes in the data map, see table T5.1 on page 59}
A2	M (Internal Relay)	Bit	0 or 1	Move Type: 0 = Relative, 1 = Absolute
A3	M (Internal Relay)	Bit	0 or 1	Move Direction: 0 = CW, 1 = CCW
A4	D (Data Register)	Word	0, 1, or 2	Acceleration Type Select 0 = Linear Acceleration 1 = Triangular S-Curve Acceleration 2 = Trapezoidal S-Curve Acceleration
A5	D (Data Register)	Word	1 to 1,000,000	Programmed Interpolated Speed
A6	D (Data Register)	Word	1 to 2000	Interpolated Acceleration in steps/millisecond/second.
A7	D (Data Register)	Word	1 to 2000	Interpolated Deceleration in steps/millisecond/second.
A8	D (Data Register)	Long†	-1,073,741,823 to +1,073,741,823	Motor Position End Point X value. (First Axis)
A9	D (Data Register)	Long†	-1,073,741,823 to +1,073,741,823	Motor Position End Point Y value. (Second Axis)
A10	D (Data Register)	Long†	-1,073,741,823 to +1,073,741,823	Move's Center Point X value. (First Axis)
A11	D (Data Register)	Long†	-1,073,741,823 to +1,073,741,823	Move's Center Point Y value. (Second Axis)

† Long integer values consume two consecutive data registers and the starting address must be even. For example, if argument A2 begins at address D49622, the next available address would be D49624. (A2 would consume both D49622 and D49623.) The starting register address of a long value must be even. (For example, argument A2 could not start with register D49621).

Table T5.22 Arguments to the "Via Point Circular Move IANF2E" Macro

5.3.22 Arguments to the "Radius Circular Move IANF2E" Macro

Note▶ Interpolated moves control both axes of the IANF2(E) simultaneously. The starting address of the command block, (argument A1), is the starting address of the first axis controlled by the IANF2(E). The command block is twenty words in length.

Argument	Device Type	Device Size	Range of Values	Description
A1	D (Data Register)	Word		Starting Address of the output registers assigned to the first axis controlled by the IANF2(E). (D49120 to D49220 in the sample programs.) {Each IANF2(E) controls two consecutive axes in the data map, see table T5.1 on page 59}
A2	M (Internal Relay)	Bit	0 or 1	Move Type: 0 = Relative, 1 = Absolute
A3	M (Internal Relay)	Bit	0 or 1	Move Direction: 0 = CW, 1 = CCW
A4	D (Data Register)	Word	0, 1, or 2	Acceleration Type Select 0 = Linear Acceleration 1 = Triangular S-Curve Acceleration 2 = Trapezoidal S-Curve Acceleration
A5	D (Data Register)	Word	1 to 1,000,000	Programmed Interpolated Speed
A6	D (Data Register)	Word	1 to 2000	Interpolated Acceleration in steps/millisecond/second.
A7	D (Data Register)	Word	1 to 2000	Interpolated Deceleration in steps/millisecond/second.
A8	D (Data Register)	Long†	-1,073,741,823 to +1,073,741,823	Motor Position End Point X value. (First Axis)
A9	D (Data Register)	Long†	-1,073,741,823 to +1,073,741,823	Motor Position End Point Y value. (Second Axis)
A10	D (Data Register)	Long†	-1,073,741,823 to +1,073,741,823	Radius

† Long integer values consume two consecutive data registers and the starting address must be even. For example, if argument A2 begins at address D49622, the next available address would be D49624. (A2 would consume both D49622 and D49623.) The starting register address of a long value must be even. (For example, argument A2 could not start with register D49621).

Table T5.23 Arguments to the "Radius Circular Move IANF2E" Macro

5.4 Add a User-defined Macro to Ladder Logic

Adding a User-defined Macro to your ladder logic requires you to set the initial conditions for the rung, select the macro, and map the macro's arguments to the Device Addresses chosen in the previous step.

5.4.1 Set initial conditions

Most macros should be triggered once to initiate a data transfer to the IANF2(E). Therefore, a **Single Output Up** instruction, (SOTU), should be added to the rung as part of the condition. The exceptions to this guideline are the Jog and Registration Moves.

5.4.2 Choose the desired macro to add to the ladder.

To add a User-defined Macro to the ladder, first, right-click where you wish to add the macro. In the popup menu that appears, hover over the **"Macro Instructions ►"** item until a second menu appears. From that menu, select **UMACRO (User-defined Macro)**. The UMACRO (User-defined Macro) dialog box will appear.

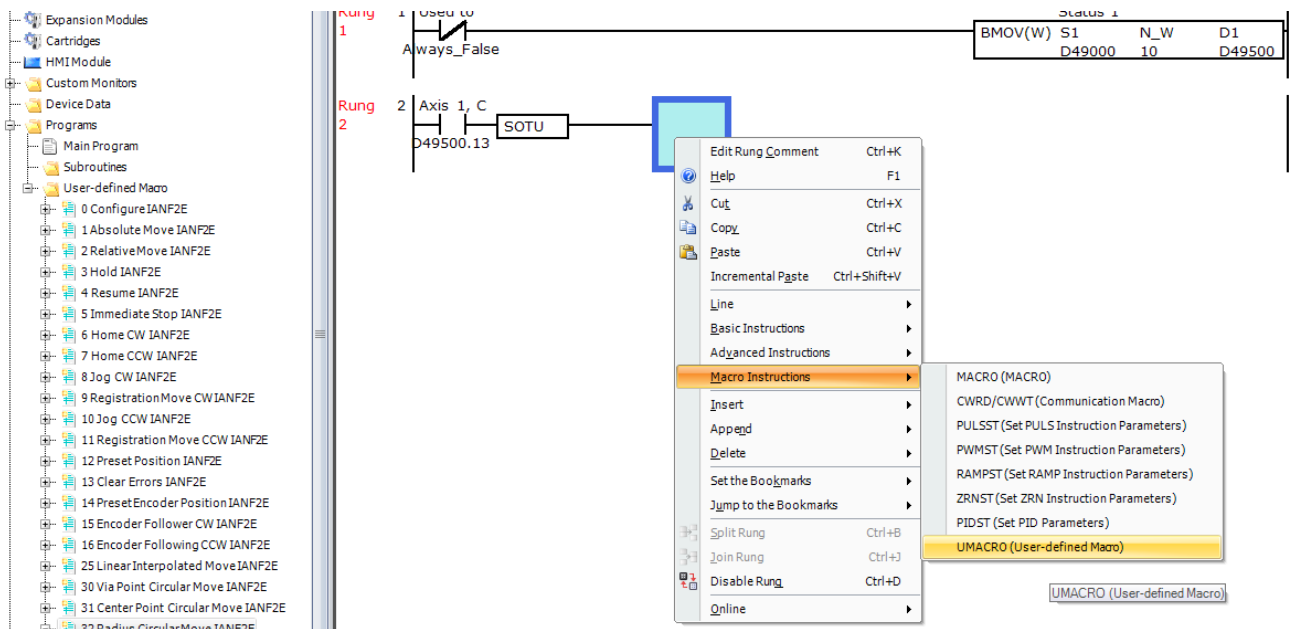


Figure T5.2 Selecting a User-defined Macro

5.4.3 Select the Proper Macro by Macro Number

Each macro is assigned a number between 0 and 255. When user-defined macros are viewed in the Project Window, these numbers are shown before the name of the macro. (See figure T5.2 above for an example.) To select the proper macro, enter the correct number in the "S1 (User-defined Macro Number):" field and press the [Tab] key. The program will open a confirmation window stating that the current settings will be lost and the argument list will be updated. Click on the [Yes] button to confirm the change.

5.4.4 Map Arguments to Device Addresses

(Continuing from the previous step.) The **UMACRO (User-defined Macro)** dialog box now displays a list of the macro's arguments along with their required Device Type. The Tag Name fields are used to map the Device Addresses to the macro's arguments. If a tag name has been defined, this name can be typed directly into the field. Optionally, the actual Device Address can be entered into the field. Finally the ellipsis button [...] can be clicked to open the **Tag Name Editor** dialog box. Once opened, it can be used to chose the desired Device Address. Figure T5.3 below shows a **UMACRO (User-defined Macro)** dialog box for the "Configure IANF2E" macro after Device Addresses have been mapped to the macro's arguments.

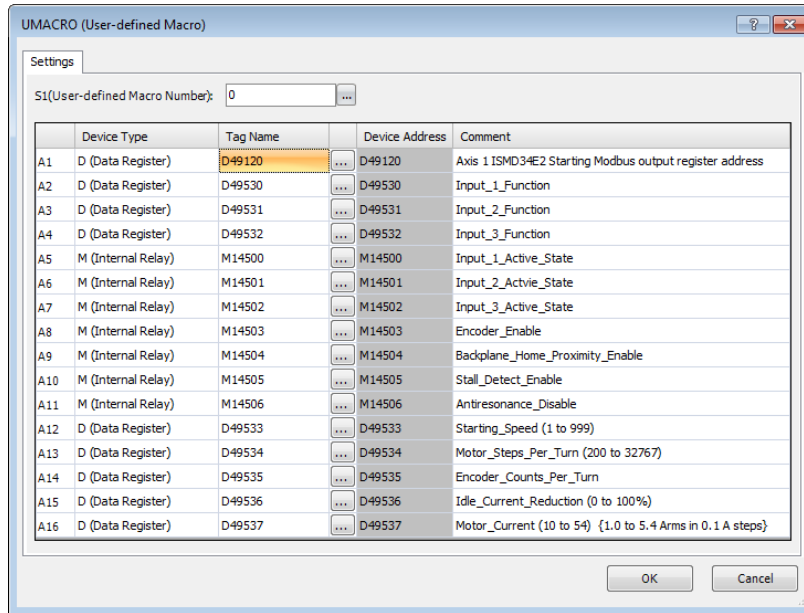


Figure T5.3 Argument Assignments

TASK 6: ADDITIONAL LOGIC AND INPUT DATA FORMATS

Introduction

The IDEC User-defined Macros simplify issuing commands and configuration data to the IANF2(E). Once a command, or configuration data, is issued, monitoring the state of the IANF2(E) is application specific and must be written. The following sections give guidelines for writing the additional logic needed to control an IANF2(E) as well as the format of the status data that is read from the IANF2(E) in both Command and Configuration modes.

6.1 Axis Independence

Except when commanding Interpolated Moves, the two axes of the IANF2(E) are completely independent. It is perfectly acceptable to have one axis in Command mode while the other is in Configuration mode.

6.2 Power-Up Behavior

The IANF2(E) will always power up in Command Mode and show a configuration error. (The first data register read from the IANF2(E) axis will have a hexadecimal value of 6408h.) Configuration data must be written down to the IANF2(E) before commands can be issued to the module.

6.3 Buffer Input Data

It is common practice to buffer the input data from the IANF2(E) to guarantee that it does not change during a program scan. IDEC sample programs use a BMOV instruction to copy data from the Modbus TCP registers to a buffer in the D49500 to D49999 range. This instruction is conditioned by an NC relay tied to a bit defined as Always_False. This buffers the data on every program scan and must occur before the data is used in the program.

6.4 Command Bits Must Transition

Note ▶ Commands are only accepted when the command bit makes a 0→1 transition. The easiest way to meet this requirement is to write a value of zero into the first of the output registers assigned to the axis of the IANF2(E) before using a macro to write out the next command. These registers are in the range of D49120 to D49239 in the IDEC sample program. See T5.1, *Reserved Memory Addresses* on page 59 for the memory layout used by the IDEC sample program.

This condition also applies when switching from Configuration Mode to Command Mode. Assume a bit is set in the first of the registers assigned to the IANF2(E) while in Configuration Mode. If a switch is made to Command Mode with the same bit set, the command will not occur because the bit must transition between writes to the unit.

6.4.1 Commands that do not cause motion: Clear Errors, preset commands, etc.

In the unlikely event that a command in this group is issued successively, its command bit must be reset and held for at least one Modbus TCP update. The time it must be held is dependent on the amount of TCP traffic in your application.

6.4.2 Jog and Registration Moves

Unlike other motion commands, the Jog and Registration Moves only occur while the command bit is set. Your program must monitor machine conditions to determine when to reset the command bit to zero. Once the command bit is off, your program should monitor the Moving_CW, Moving_CCW, or Stopped status bits to verify that the axis has stopped moving before issuing another command.

6.4.3 Interpolated Moves, Absolute Move, Relative Move, and Find Home

Once issued, these commands will run to their completion regardless of the state of their command bit(s). Once the command has been issued, your program can monitor the Moving_CW, Moving_CCW, or Stopped status bits to verify the axis is in motion and reset the command word at that point. Unless the commanded move is very short, a network update will occur while motion is occurring and the IANF2(E) will see the reset command bit. As with the Jog and Registration Moves, the program should monitor the Moving_CW, Moving_CCW, or Stopped status bits to verify that the axis has stopped moving before issuing another command.

6.5 Command Mode Input Data Format

The format for the Network Input Data when an IANF2(E) axis is in Command Mode is shown below. Both axes are shown in command mode.

Offset From First IANF2(E) Read Register Address ⁽¹⁾	Axis	Command Mode Input Data
0	1	Status Word 0 ⁽²⁾
1	1	Status Word 1
2	1	32 bit Motor Position: Upper 16 bits
3	1	32 bit Motor Position: Lower 16 bits
4	1	32 bit Encoder Position ⁽³⁾ : Upper 16 bits
5	1	32 bit Encoder Position ⁽³⁾ : Lower 16 bits
6	1	32 bit Captured Position ⁽⁴⁾ : Upper 16 bits
7	1	32 bit Captured Position ⁽⁴⁾ : Lower 16 bits
8	–	Reserved
9	–	Network Connectivity
10	2	Status Word 0 ⁽²⁾
11	2	Status Word 1
12	2	32 bit Motor Position: Upper 16 bits
13	2	32 bit Motor Position: Lower 16 bits
14	2	32 bit Encoder Position ⁽³⁾ : Upper 16 bits
15	2	32 bit Encoder Position ⁽³⁾ : Lower 16 bits
16	2	32 bit Captured Position ⁽⁴⁾ : Upper 16 bits
17	2	32 bit Captured Position ⁽⁴⁾ : Lower 16 bits
18	–	Reserved
19	–	Reserved

(1) When using the User-defined Macros, the first Modbus Read Register address is mapped into the range of registers D49000 to D49100. See table T5.1, *Reserved Memory Addresses*, on page 59 for a list of the addresses for each axis.

(2) Status Word 0, bit 15 equals zero when the axis is in Command mode.

(3) These registers contain the 32 bit encoder value when the encoder inputs are enabled. They contain the Diagnostic Feedback Value when the axis is configured for diagnostic feedback.

(4) When the encoder inputs for the axis are enabled, the IANF2(E) will capture the present encoder position when the Capture Input makes an inactive-to-active transition. If the encoder inputs are not enabled, the IANF2(E) will capture the present motor position.

Table T6.1 Command Mode Input Data Format

6.5.1 Status Word 0

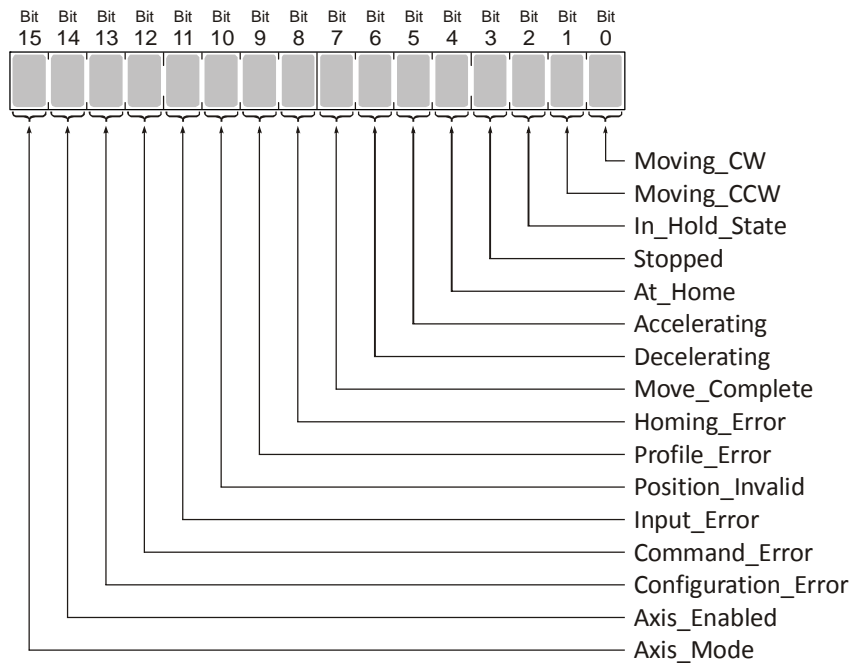


Figure T6.1 Command Mode: Status Word 0

- Bit 0: Moving_CW** – Set to “1” when the IANF2(E) is outputting clockwise pulses, otherwise “0”.
- Bit 1: Moving_CCW** – Set to “1” when the IANF2(E) is outputting counter-clockwise pulses, otherwise “0”.
- Bit 2: In_Hold_State** – Set to “1” when a move command has been successfully brought into a Hold State. Hold States are explained in the *Controlling Moves In Progress* section starting on page 25.
- Bit 3: Stopped** – Set to “1” when the pulse train output is not changing. Note that this is stopped for any reason, not just a completed move. For example, an Immediate Stop command during a move will set this bit to “1”, but the Move_Complete Bit, (bit 7 below) will not be set.
- Bit 4: At_Home** – Set to “1” when a homing command has completed successfully, “0” at all other times. This bit is only set at the successful completion of a Find Home command. This bit is not set if the home position is reached by any other move.
- Bit 5: Accelerating** – Set to “1” when the present move is accelerating. Set to “0” at all other times.
- Bit 6: Decelerating** – Set to “1” when the present move is decelerating. Set to “0” at all other times.
- Bit 7: Move_Complete** – Set to “1” when the axis has successfully completed an Absolute, Relative, Blend, or Interpolated Move. An Interpolated Move will set the Move_Complete bits of both axes. This bit is reset to “0” when the next move command is written to the IANF2(E), or when the position is preset. This bit is not set at the end of a homing operation.
- Bit 8: Homing_Error** – Set when the IANF2(E) could not home the axis because of an error in the Home CW/CCW command data, the homing operation exceeded its programmed timeout value, an E-Stop input activates, or the limit switch that is opposite to the current direction of travel activates.
- Bit 9: Profile_Error** – Set under the following conditions:
- Error in changing parameters during a Jog or Registration Move.
 - There was an error in the last Program Blend Profile data block written to the IANF2(E) axis. Currently, User-defined Macros do not exist for programming Blend Moves. See the reference section *Blend Moves*, starting on page 105, for information on programming Blend Moves.
- Bit 10: Position_Invalid** – Set under the following conditions:
- Axis switched from Configuration Mode to Command Mode
 - An Immediate Stop command was issued to the axis while motion was occurring
 - An Emergency Stop input was activated while motion was occurring.
 - If either of the CW or CCW Limit inputs become active during a Relative, Absolute, Blend, or Interpolated Move.

Bit 11: Input_Error –

Home CW/CCW Commands: Set when an E-Stop input activates during homing operation or when the limit switch that is opposite to the direction of travel activates, for example when the CCW Limit activates during a +Find Home command.

Jog CW/CCW Commands: Set when an E-Stop input activates during the move or when the limit switch that is in the direction of travel activates, for example when the CW Limit activates during a +Jog Move command.

Registration Move CW/CCW Commands: Set when an E-Stop input activates during the move or when the limit switch that is in the direction of travel activates, for example when the CW Limit activates during a +Registration Move command.

All Other Move Commands: Set when an E-Stop input activates during the move or when either of the end limit switches activates.

Note that the Input Error bit is latched but you do not have to issue a Clear Errors command block to clear the bit. The next move command will clear this bit automatically if none of the inputs are active when the command is issued.

Bit 12: Command_Error – Set when the last command issued to the IANF2(E) axis forced an error. When there is an error in an Interpolated Move, only the Command Error bit for axis 1 is set. This bit must be reset by issuing a Reset Errors command to the IANF2(E) axis. Other commands, including the Immediate Stop command, will not be accepted until this error is reset.

Bit 13: Configuration_Error – Set to “1” when the axis has a configuration error.

Bit 14: Axis_Enabled – Set to “1” when the axis is enabled. Axis 2 of an IANF2(E) is disabled by default. An axis is automatically enabled when valid configuration data is written to it.

Bit 15: Axis_Mode – Set to “1” when the axis is in Configuration Mode. Reset to “0” when the axis is in Command Mode.

6.5.2 Status Word 1

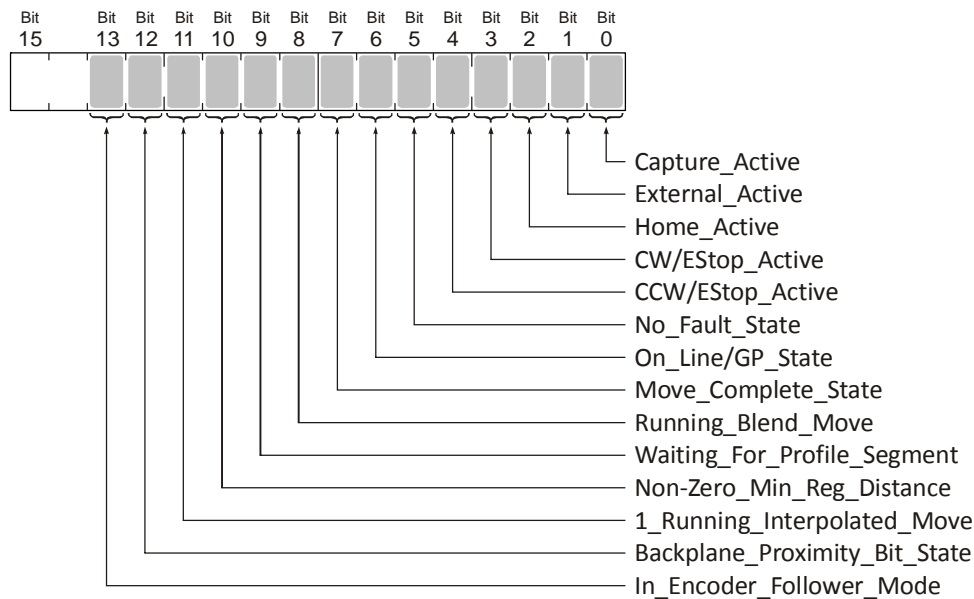


Figure T6.2 Command Mode: Status Word 1

Bit 0: Capture_Active – “1” when the Capture Input is enabled and in its active state. The enabled/disabled state of the input is programmed with argument A2 of the “Configure IANF2E” macro. The active state of the input is programmed with argument A3 of the macro. *Arguments to the “Configure IANF2” Macro* can be found on page 61.

Bit 1: External_Active – “1” when the External Input is enabled and in its active state. The enabled/disabled state of the input is programmed with argument A4 of the “Configure IANF2E” macro. The active state of the input is programmed with argument A5 of the macro. *Arguments to the “Configure IANF2” Macro* can be found on page 61.

Bit 2: Home_Active – “1” when the External Input is enabled and in its active state. The enabled/disabled state of the input is programmed with argument A6 of the “Configure IANF2E” macro. The active state of the input is programmed with argument A7 of the macro. *Arguments to the “Configure IANF2” Macro* can be found on page 61.

Bit 3: CW/Estop_Active – “1” when the CW/Estop is enabled and in its active state. The enabled/disabled state of the input is programmed with argument A8 of the “Configure IANF2E” macro. The active state of the input is programmed with argument A9 of the macro. *Arguments to the “Configure IANF2” Macro* can be found on page 61.

Bit 4: CCW/Estop_Active – “1” when the CCW/Estop is enabled and in its active state. The enabled/disabled state of the input is programmed with argument A10 of the “Configure IANF2E” macro. The active state of the input is programmed with argument A11 of the macro. *Arguments to the “Configure IANF2” Macro* can be found on page 61.

Bit 5: No_Fault_State – This bit shows the state of the No_Fault output for the axis. This bit must be set to its active state (“1”) by the IDEC controller. Once set, this bit will be reset by the following conditions:

- Input Error during a move. (E-Stop, CW Limit, or CCW Limit Input activated during move.)
- Axis switched from Command Mode to Configuration Mode
- Loss of network communications
- Reset by host controller.

Bit 6: On_Line/GP_State: This bit shows the state of the On_Line output for axis 1 and the state of the General Purpose output for axis 2. The On_Line bit must be set to its active state (“1”) by the host controller. Once set, this bit will be reset by the following conditions:

- Loss of network communications
- Reset by host controller.

The state of the General Purpose output is only controlled by the host controller.

Bit 7: Move_Complete_State: This bit shows the state of the Move Complete output for the axis. This bit will be set by the following conditions:

- Absolute, Relative, Blend, or Interpolated Move completes without error.
- Set by host controller.

Once set, this bit must be reset to its inactive state (“0”) by the host controller.

Bit 8: Running_Blend_Move: Set when the axis is presently running a Blend Move Profile. This bit is reset when the move completes. Programming and running Blend Moves is discussed in the reference section *Blend Moves*, starting on page 105.

Bit 9: Waiting_For_Profile_Segment: Set when IANF2(E) has accepted a Blend Move Profile programming block (0→1 transition) and when it is ready to accept the next (1→0 transition). See *Programming Blend Move Profiles*, which starts on page 106, for additional information.

Bit 10: Non-Zero_Min_Reg_Distance: Set when the Minimum Registration Move Distance parameter is programmed to a non-zero value.

Bit 11: 1_Running_Interpolated_Move: Set in axis 1 status bits when the IANF2(E) is running an Interpolated Move. This bit is reset when the move comes to an end. For axis 2, this bit will always equal zero.

Bit 12: Backplane_Proximity_Bit_State: This bit always equals the state of the Backplane_Proximity_Bit, which is bit 6 in the *Command Word 1* for the axis.

Bit 13: Encoder Follower Mode: Set when the axis is in Encoder Follower Mode.

Bits 14 & 15: Reserved: These bits will always equal zero.

6.5.3 Network Connectivity (Word 9)

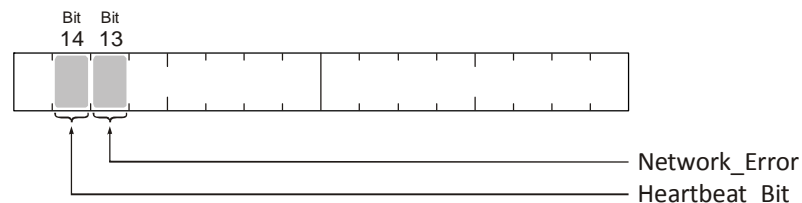


Figure T6.3 Network Connectivity Status Word

Bits 1 - 12: Backplane_Proximity_Bit_State: Reserved: These bits will always equal zero.

Bit 13: Network_Error: This bit is set to “1” if the module loses its network connection. This bit is set until a Reset Errors command is sent to the module. All other commands will not be accepted until this error is cleared.

Bit 13: Heartbeat_Bit: This bit will change state approximately every 500 milliseconds. Monitor this bit to verify that the unit and network connection are operating correctly.

Bit 15: Reserved: This bit will always equal zero.

6.6 Configuration Mode Input Data Format

The format for the Network Input Data when an IANF2(E) axis is in Configuration Mode is shown below. Both axes are shown in Configuration Mode, but it is valid to have one axis in Configuration Mode while the other is in Command Mode.

Offset From First IANF2(E) Read Register Address ⁽¹⁾	Axis	Command Mode Input Data
0	1	Bits 0 - 12: Mirror of Configuration Word 0 ⁽²⁾⁽³⁾
1	1	Mirror of Configuration Word 1
2	1	<i>Not used in Configuration Mode, will always be zero.</i>
3	1	Mirror of configured Starting Speed
4	1	Mirror of configured Homing Timeout value.
5	1	<i>Not used in Configuration Mode, will always be zero.</i>
6	1	<i>Not used in Configuration Mode, will always be zero.</i>
7	1	<i>Not used in Configuration Mode, will always be zero.</i>
8	1	<i>Not used in Configuration Mode, will always be zero.</i>
9	1	Network Connectivity (See 6.5.3 on page 77)
10	2	Bits 0 - 12: Mirror of Configuration Word 0 ⁽²⁾⁽³⁾
11	2	Configuration Word 1
12	2	<i>Not used in Configuration Mode, will always be zero.</i>
13	2	Mirror of configured Starting Speed
14	2	Mirror of configured Homing Timeout value.
15	2	<i>Not used in Configuration Mode, will always be zero.</i>
16	2	<i>Not used in Configuration Mode, will always be zero.</i>
17	2	<i>Not used in Configuration Mode, will always be zero.</i>
18	2	<i>Not used in Configuration Mode, will always be zero.</i>
19	2	<i>Not used in Configuration Mode, will always be zero.</i>

(1) When using the User-defined Macros, the first Modbus Read Register address is mapped into the range of registers D49000 to D49100. See table T5.1, *Reserved Memory Addresses*, on page 59 for a list of the addresses for each axis.

(2) Bits 15, 14, and 13 have the same meaning when in Command Mode. These three bits are used to show the state of the axis and are not mirrored.

(3) Bit 15 of Configuration Word 0 equals "1" when in Configuration Mode.

Figure T6.4 Configuration Mode Input Data Format

6.6.1 Configuration Word 0 Mirror

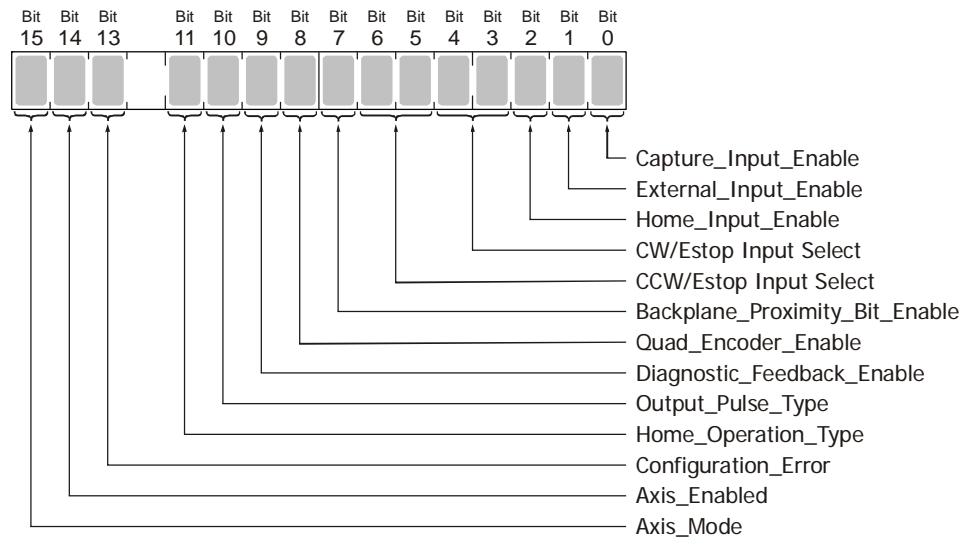


Figure R5.6 Configuration Mode: Status Word 0

Bit 0: Capture_Input_Enable – Set to “1” when the Capture Input is enabled.

Bit 1: External_Input_Enable – Set to “1” when the External Input is enabled.

Bit 2: Home_Input_Enable – Set to “1” when the Home Input is enabled.

Bits 3&4: CW/Estop Input Select –

Bit 4	Bit 3	Description
0	0	Disabled
0	1	CW Limit Switch
1	0	E-Stop Input
1	1	Invalid Combination

Bits 5&6: CCW/Estop Input Select –

Bit 4	Bit 3	Description
0	0	Disabled
0	1	CCW Limit Switch
1	0	E-Stop Input
1	1	Invalid Combination

Bit 7: Backplane_Proximity_Bit_Enable – Set to “1” when the Backplane Proximity Bit is enabled.

Bit 8: Quad_Encoder_Enable – Set to “1” when the quadrature encoder inputs are enabled.

Bit 9: Diagnostic_Feedback_Enable – Set to “1” when diagnostic feedback mode is enabled.

Bit 10: Output_Pulse_Type – Set to “0” when the axis outputs pulses in the CW/CCW format. Set to “1” when the axis outputs pulses in the Step/Direction format.

Bit 11: Home_Operation_Type – Set to “0” when the axis will home to the Home Limit Switch. Set to “1” when the axis will home to the quadrature encoder’s marker pulse.

Bit 12: Reserved – Will always equal “0”.

Bit 13: Configuration_Error – Set to “1” when the axis has a configuration error.

Bit 14: Axis_Enabled – Set to “1” when the axis is enabled. Axis 2 of an IANF2(E) is disabled by default. An axis is automatically enabled when valid configuration data is written to it.

Bit 15: Axis_Mode – Set to “1” when the axis is in Configuration Mode. Reset to “0” when the axis is in Command Mode.

6.6.2 Configuration Word 1 Mirror

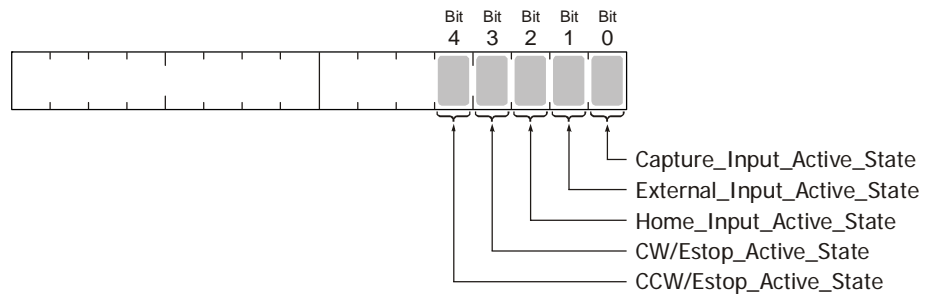


Figure R5.7 Configuration Mode: Status Word 1

- Bit 0: Capture_Input_Active_State** – “0” = NC Input, “1” = NO Input. NC inputs are active when power is *not* applied.
- Bit 1: External_Input_Active_State** – “0” = NC Input, “1” = NO Input. NC inputs are active when power is *not* applied.
- Bit 2: Home_Input_Active_State** – “0” = NC Input, “1” = NO Input. NC inputs are active when power is *not* applied.
- Bit 3: CW/Estop_Active_State** – “0” = NC Input, “1” = NO Input. NC inputs are active when power is *not* applied.
- Bit 4: CCW/Estop_Active_State** – “0” = NC Input, “1” = NO Input. NC inputs are active when power is *not* applied.

REFERENCE 6: TABULATED COMMAND AND ERROR CODES

Introduction

This chapter is broken down into two sections. The first section lists status bits and the conditions that will set the bit. You can assume that the bit will be reset under all conditions that are not listed. The second section lists the available commands and the errors that can occur with them.

Status Bits

Axis Stopped

Axis 1: Input Word 0, bit 3 Axis 2: Input Word 10, bit 3

- 1) Set whenever the axis is in command mode and motion is not occurring.

Move Complete Bit

Axis 1: Input Word 0, bit 7 Axis 2: Input Word 10, bit 7

- 1) Set after an Absolute, Relative, Blend, or Interpolated Move has been successively completed. The Move Complete bit will be reset when the next move operation is initiated or by a Preset command. This bit is also reset when switching from Configuration Mode to Command Mode.

Home Invalid Bit

Axis 1: Input Word 0, bit 8 Axis 2: Input Word 10, bit 8

The Home Invalid bit will be set under the following conditions:

- 1) Issuing a Home CW or Home CCW command with the Programmed Speed less than the configured Starting Speed or greater than 1,000,000 steps/second.
- 2) Issuing a Home CW or Home CCW command using either of the S-Curve Acceleration or Deceleration Types. (Only *Constant Acceleration* is available with Home CW/CCW commands.
- 3) Issuing a Home CW or Home CCW command while having an acceleration or deceleration values outside their valid range of 1 to 2000.
- 4) Issuing a Homing Command with any reserved words not equal to zero.
- 5) If the opposite End Limit Switch is reached during a homing operation. For example, if the CCW Limit Switch is reached during a Home CW operation. Improper wiring, configuration, or placement of the limit switch would most likely cause this error.
- 6) If the CW Limit or CCW Limit activate at the same time as the Home Limit.
- 7) If the time of the homing operation exceeds the configured Homing Timeout value.

Profile Invalid Bit

Axis1: Input Word 0, bit 9 Axis 2: Input Word 10, bit 9

The Invalid Profile bit will be set under the following conditions

- 1) If the Programmed Speed of a Relative, Absolute, Blend, or Interpolated Move is less than the configured Starting Speed
- 2) If any of the move parameters are outside of their valid ranges.
- 3) If the defined profile overflows the internal registers of the IANF2(E) module.
- 4) Setting the number of Blend Move Segments to be less than 2 or greater than 16.
- 5) If a Blend Move segment does not reach the Programmed Speed. That is, if the distance that it takes to accelerate to the programmed speed exceeds the length of the segment.
- 6) Two consecutive Blend Move segments with the same Programmed Speed.
- 7) Blend Move profile Initial Profile Programming Block that specifies an Acceleration Type of "10" in any of the bit pair in words 3, 4, & 5 (axis1) or 11, 12, & 13 (axis2)
- 8) Changing the speed of a Jog CW or Jog CCW command whose initial speed was less than or equal to the configured Starting Speed.
- 9) If invalid changes are made to the velocity, acceleration, or deceleration parameters of a Jog CW or Jog CCW command while it is occurring. When detected, motion will immediately stop.
- 10) Motion will immediately stop if a non-zero value is entered into the Reserved words (Axis1: 2/3, Axis2: 10/11), while a Jog CW or Jog CCW move is occurring.
- 11) If the Minimum Registration Distance parameter equals zero and a Registration Move is started while the External Input active.
- 12) Starting a Registration Move CW or Registration Move CCW command if the number of steps programmed into the Position Offset register is less than the number of steps required to decelerate and stop the motion. In this case the Registration Move will not start.

- 13) If the Position Offset of a Registration Move CW or Registration Move CCW command is negative.
- 14) If the value of the Position Offset registers is changed after a Registration Move CW or Registration Move CCW command has been initiated. Motion will immediately stop.
- 15) Increasing the speed of a a Registration Move CW or Registration Move CCW while it is occurring. Motion will immediately stop. Please note that it is possible to decrease the speed of Registration Move while it is occurring.
- 16) Changing the acceleration or deceleration parameters during a Registration Move, however, these values are only checked when the Programmed Speed is changed. Motion will immediately stop.
- 17) Data in reserved registers during an Interpolated Move.

Position Invalid Bit

Axis 1: Input Word 0, bit 10 Axis 2: Input Word 10, bit 10

The Position Invalid will be set under the following conditions

- 1) When switching from Configuration Mode to Command Mode. This will be true even if the position was valid before the configuration operation occurred.
- 2) After an Immediate Stop command has been issued.
- 3) If the Emergency Stop input has been activated.
- 4) If either of the CW Limit or CCW Limit inputs become active during a Relative, Absolute, Blend, Interpolated or Encoder Follower Move.
- 5) If the limit input that is the same as the commanded direction of travel is reached during a Jog or Registration Move. For example, if the CW Limit Switch is reached during a +Jog Move.
- 6) If the limit input that is the opposite of the commanded direction of travel is reached during a homing operation. For example, if the CCW Limit Switch is reached during a +Find Home command.

Input Error Bit

Axis1: Input Word 0, bit 11 Axis 2: Input Word 10, bit 11

The Input Error bit will be set under the following conditions

- 1) If an Emergency Stop input is activated.
- 2) If either of the CW Limit or CCW Limit inputs become active during a Relative, Absolute, Blend, Encoder Follower, or Interpolated Move.
- 3) If the limit input that is the same as the commanded direction of travel is reached during a Jog or Registration Move. For example, if the CW Limit Switch is reached during a +Jog Move.
- 4) If the limit input that is the opposite of the commanded direction of travel is reached during a homing operation. For example, if the CCW Limit Switch is reached during a +Find Home command.
- 5) Performing a Jog Move in the same direction as the active End Limit Switch. For example, issuing a Jog CW command when the CW Limit Switch is active.

Command Error Bit

Axis 1: Input Word 0, bit 12 Axis 2: Input Word 10, bit 12

The Command Error bit will be set under the following conditions

- 1) Attempting to enter Configuration Mode while a move operation is occurring on the axis. The move will run to completion and the axis will then enter Configuration Mode, clearing the error.
- 2) If more than one bit in the Command Bits MSW is set at one time. (Axis1: Output Word 0, Axis2: Output Word 10)
- 3) If any bit in the Command Bits MSW word is set while also commanding an Interpolated Move with the Command Bits LSW bits 9 through 14.
- 4) If a move operation is started before the previous move operation is completed. The present move will run to completion.
- 5) If the Target Position, Programmed Speed, Acceleration, or Deceleration parameters are outside of their valid ranges.
- 6) If the Programmed Speed is less than the configured Starting Speed for Absolute, Relative, Blend, and Interpolated Moves as well as Find Home commands. Jog and Registration Moves can be run at speeds less than the starting speed.
- 7) Performing an Absolute Move or an Absolute Interpolated Move when the Current Position is not valid.
- 8) Issuing a Hold Move command while a Blend, Registration, or Find Home command is running.
- 9) Issuing a Resume Move command when the axis is not in a Hold state.
- 10) Issuing any command with the Reserved words in the command having a non-zero value.
- 11) Issuing a Find Home command with S-Curve Acceleration selected.
- 12) Issuing a Find Home command before configuring a homing input. (Home Input or Z-pulse)
- 13) Issuing a Preset Encoder command when axis in not configured to use an encoder.

*Configuration Error Bit***Axis1: Input Word 0, bit 13 Axis 2: Input Word 10, bit 13**

The Configuration Error bit will be set under the following conditions

- 1) At power up, before the IANF2(E) has been configured for the first time.
- 2) If an invalid configuration has been written to the axis.

Command Error Conditions By Command

This section list the errors that will cause the Command Error bit to be set. The Command Error bit is Input Word 0, bit 12 for axis 1 and Input Word 10, bit 12 for axis 2.

Jog CW/CCW Command

- 1) Changing the speed of a Jog Move whose initial speed was less than or equal to the configured Starting Speed.
- 2) If invalid changes are made to the velocity, acceleration, or deceleration parameters while the move is occurring. When detected, motion will immediately stop.
- 3) If a non-zero value is entered in the Reserved registers while a Jog Move is occurring. Motion will immediately stop. If a non-zero value is entered when starting a Jog Move, the IANF2(E) axis will perform a Registration Move instead.

Blend CW/CCW Command

- 1) If a Blend Move command is initiated before the profile was programmed for the axis.
- 2) Programming a Blend Move Profile without a final deceleration value. This parameter is programmed in word 5 of the Initial Profile Programming Block.
- 3) If a Blend Move command is sent to the axis with any of the Reserved words not set to zero.
- 4) If a blend move profile is programmed while a blend move operation is in progress. If this occurs, the deceleration value of the final segment will be used to bring the move to a Controlled Stop.
- 5) If the Hold Move command is issued while a blend move profile is running. If this occurs, the command error bit will be set while the blend move profile runs to completion. When completed, the command error will be reset.

Registration Move Command

- 1) If the Minimum Registration Distance is outside of the valid range or is negative.
- 2) If the Hold Command bit is set while a Registration Move is occurring.
- 3) If the Registration Move is started with the External Input active.
- 4) If the number of steps programmed into the Position Offset registers during a Registration Move is less than the number of steps required to decelerate and stop the motion. In this case the motion will not start.
- 5) If the number of steps programmed into the Position Offset registers is negative.
- 6) If the value of the Position Offset registers is changed after a Registration Move has been initiated. Motion will immediately stop.
- 7) If the Registration Move speed is increased while motion is occurring. Motion will immediately stop. Please note that it is allowed to decrease the speed while a Registration Move is occurring.
- 8) Changing the acceleration or deceleration parameters. These values are only checked when the Programmed Speed is changed. Motion will immediately stop.

Encoder Follower Operations

- 1) If the IANF2(E) axis is placed in Encoder Follower mode, without first configuring the axis to use an encoder.
- 2) Trying to place the axis in Encoder Follower mode while there is an active Move command running. Please note that even though the command error status bit will be set, the original move operation will run to completion. You can switch to Encoder Follower Mode while motion is occurring on the encoder inputs.
- 3) If the Multiplier or Divisor values are outside of the range of 1 to 32767.
- 4) If either Reserved words are not equal to zero.
- 5) Attempt to change the direction while there is motion on the axis. The Axis Stopped bit (Axis1: I:X.0/3, Axis2: I:X.8/3) must be set before the direction can be changed. You must also end the first Encoder Follower command by resetting the Command Bits MSW word to zero before issuing the Encoder Follower command that reverses direction.
- 6) If any command is issued, including the Hold Command, when the axis module is in Encoder Follower mode. That is, if any of the bits in the Command Bits MSW other than Jog_CW or Jog_CCW bits are set when the Encoder Follower Bit is also set.
- 7) If the Encoder Follower Bit is reset while the Jog CW/CCW bits remain set during an Encoder Follower operation.
- 8) Selecting S curve acceleration with Encoder Follower mode.

Notes

REFERENCE 7: CONFIG MODE OUTPUT DATA FORMAT

The custom macros supplied by IDEC simplify the configuration and programming of the IANF2(E) units. The data in this reference is presented as background information to aid in troubleshooting.

Introduction

This chapter covers the format of the Network Output Data used to configure the IANF2(E). An IANF2(E) requires twenty data registers for output data to the module as well as twenty data registers to hold data read from the module. When configuring the IANF2(E) the first ten words configures the first axis, while the second ten words configures the second axis. The format of this data is a mix of sixteen bit and thirty-two bit integers.

Power Up Behavior

The IANF2(E) will always power up with both axes in Command Mode and report a configuration error. An axis must be configured before you can issue move commands to the axis. Note that the two axes are independent, and it is acceptable to have one axis in Configuration Mode while the other is in Command Mode. However, both axes must be properly configured and in Command Mode before you can issue Interpolated Move commands.

Data Format

An IANF2(E) requires twenty registers for output data as well as twenty registers for input data. These words are broken down into a combination of sixteen and thirty-two bit integers.

The Starting Speed is a thirty-two bit value and is transmitted most significant word first (big endian). This is the data format specified by the Modbus specification and used by IDEC controllers. Table R7.1 below shows the format of the data as it is stored and transmitted.

Value	32 bit Integer Format	
	First Word	Second Word
12	0x0000	0x000C
1,000,000	0x000F	0x4240

Table R7.1 Position Data Format Examples

Output Data Format

The following table shows the format of the output network data words when writing configuration data to both axes of the IANF2(E).

Offset From First IANF2(E) Write Register Address ⁽¹⁾	Axis	Command Mode Input Data
0	1	Configuration Word 0 ⁽²⁾
1	1	Configuration Word 1
2	1	<i>Not used in Configuration Mode, must be set to zero.</i>
3	1	Starting Speed
4	1	Homing Timeout value.
5	1	<i>Not used in Configuration Mode, must be set to zero.</i>
6	1	<i>Not used in Configuration Mode, must be set to zero.</i>
7	1	<i>Not used in Configuration Mode, must be set to zero.</i>
8	1	<i>Not used in Configuration Mode, must be set to zero.</i>
9	1	<i>Not used in Configuration Mode, must be set to zero.</i>
10	2	Configuration Word 0 ⁽²⁾
11	2	Configuration Word 1
12	2	<i>Not used in Configuration Mode, must be set to zero.</i>
13	2	Starting Speed
14	2	Homing Timeout value.
15	2	<i>Not used in Configuration Mode, must be set to zero.</i>
16	2	<i>Not used in Configuration Mode, must be set to zero.</i>
17	2	<i>Not used in Configuration Mode, must be set to zero.</i>
18	2	<i>Not used in Configuration Mode, must be set to zero.</i>
19	2	<i>Not used in Configuration Mode, must be set to zero.</i>

(1) When using the User-defined Macros, the first Modbus Write Register address is mapped into the range of registers D49120 to D49220. See table T5.1, *Reserved Memory Addresses*, on page 59 for a list of the addresses for each axis.

(2) Bit 15 of Configuration Word 0 must equal "1" when in Configuration Mode.

Figure T6.5 Output Data Format - Configuration Data

When configuration data is written to the axis, the IANF2(E) will respond in one of two ways:

- If the configuration data is invalid, the IANF2(E) will stay in Command Mode and will set the Configuration_Error bit.
- If the configuration data is valid, the IANF2(E) will switch the axis to Configuration Mode and mirror the accepted data. The format of the mirrored data is given in section 6.6 *Configuration Mode Input Data Format* starting on page 78.

Configuration Word 0

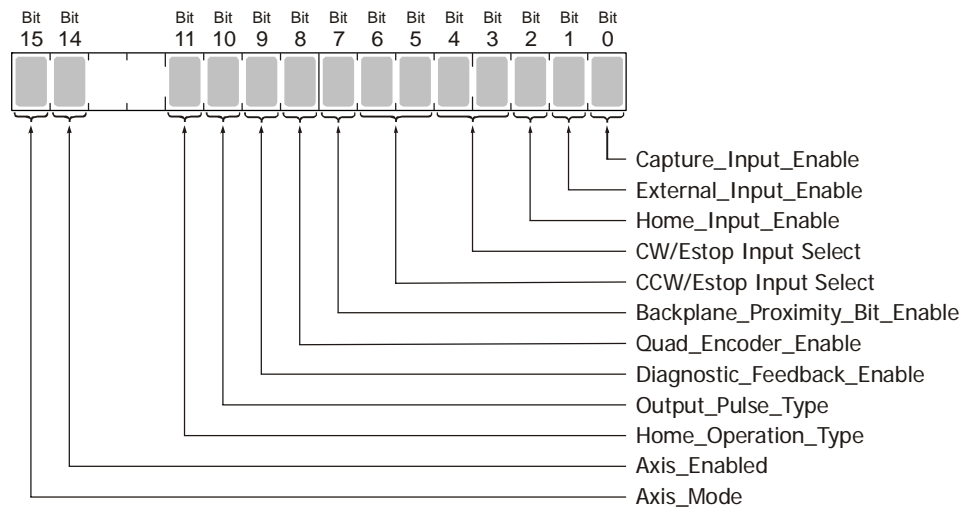


Figure R7.1 Configuration Mode: Status Word 0

Bit 0: Capture_Input_Enable – Set to “1” to enable the Capture Input. Reset to “0” to disable the input.

Bit 1: External_Input_Enable – Set to “1” to enable the External Input. Reset to “0” to disable the input.

Bit 2: Home_Input_Enable – Set to “1” to enable the Home Input. Reset to “0” to disable the input.

Bits 3&4: CW/Estop Input Select –

Bit 4	Bit 3	Description
0	0	Disable
0	1	CW Limit Switch
1	0	E-Stop Input
1	1	Invalid Combination

Bits 5&6: CCW/Estop Input Select –

Bit 4	Bit 3	Description
0	0	Disable
0	1	CCW Limit Switch
1	0	E-Stop Input
1	1	Invalid Combination

Bit 7: Backplane_Proximity_Bit_Enable – Set to “1” to enable the Backplane Proximity Bit. Reset to “0” to disable.

Bit 8: Quad_Encoder_Enable – Set to “1” to enable the quadrature encoder inputs.

Bit 9: Diagnostic_Feedback_Enable – Set to “1” to enable the diagnostic feedback mode Reset to “0” for normal operation.

Bit 10: Output_Pulse_Type – Set to “0” to configure the axis to output pulses in the CW/CCW format. Set to “1” to configure the axis to output pulses in the Step/Direction format.

Bit 11: Home_Operation_Type – Set to “0” when the axis will home to the Home Limit Switch. Set to “1” when the axis will home to the quadrature encoder’s marker pulse.

Bits 12 & 13: Reserved – Must reset to “0”.

Bit 14: Axis_Enabled – Set to “1” to enable the axis. Reset to “0” to disable the axis. Axis 2 of an IANF2(E) is disabled by default. An axis is automatically enabled when valid configuration data is written to it.

Bit 15: Axis_Mode – Set to “1” when the axis is in Configuration Mode. Reset to “0” when the axis is in Command Mode.

6.6.3 Configuration Word 1

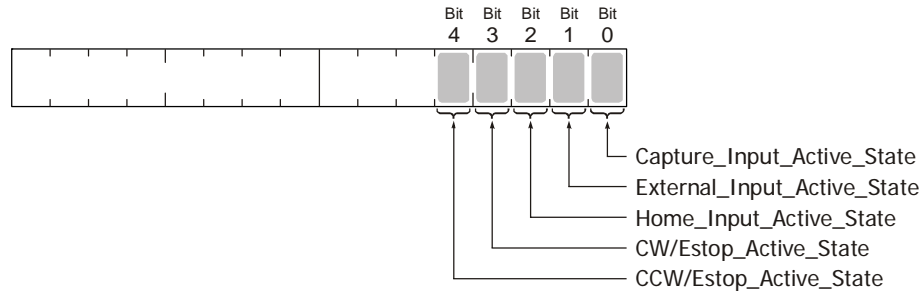


Figure R7.2 Configuration Mode: Status Word 1

- Bit 0: Capture_Input_Active_State** – “0” = NC Input, “1” = NO Input. NC inputs are active when power is *not* applied.
- Bit 1: External_Input_Active_State** – “0” = NC Input, “1” = NO Input. NC inputs are active when power is *not* applied.
- Bit 2: Home_Input_Active_State** – “0” = NC Input, “1” = NO Input. NC inputs are active when power is *not* applied.
- Bit 3: CW/Estop_Active_State** – “0” = NC Input, “1” = NO Input. NC inputs are active when power is *not* applied.
- Bit 4: CCW/Estop_Active_State** – “0” = NC Input, “1” = NO Input. NC inputs are active when power is *not* applied.

Starting Speed

The Starting Speed parameter sets the step frequency that all moves begin and end at. Portions of a homing operation are also performed at this speed. This parameter is used to avoid any low frequency resonance points when starting a move. This parameter is also useful in micro-stepping applications when it is not necessary to accelerate from a full stop. The range of the Starting Speed parameter is 1 to 1,000,000 steps per second.

As a suggestion, a Starting Speed that is equal to the square root of the acceleration rate will create a profile without any jumps in velocity at the beginning of the move profile. The same holds true for the end of the move profile if the Starting Speed is equal to the square root of the deceleration rate.

Homing Timeout

The Homing Timeout parameter defines how long a homing operation will be allowed to continue before motion stops and the IANF2(E) sets its *Home Invalid* status bit. (The *Home Invalid* status bit is available when the axis is in Command Mode.) The Homing Timeout parameter has a range of 0 to 300 seconds. Setting this parameter to a value of 0 disables this feature and the IANF2(E) will search for the home marker until it finds it or the command is aborted by the IDEC controller.

Invalid Configurations

The following section lists the configuration settings that will cause a configuration error on the axis. Input word 0, bit 13 is the Configuration Error bit for axis 1 and input word 10, bit 13 is the Configuration Error bit for axis 2.

In the following tables, all bit places that do not show a value are “don’t care” terms for that error.

Configuration Word 0

Word 0 configures Axis 1, Word 10 configures Axis 2

If these bits are set as shown....																...a Configuration Error will result because:
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
NOTE: This bit is always set when in Configuration Mode			1	1												ANY of the reserved bits are set
												1	1			Configuration tried to use an invalid setting for the CW/E-Stop Input.
										1	1					Configuration tried to use an invalid setting for the CCW/E-Stop Input.
						0	1									Configuration tried to enable Diagnostic Feedback while the Output Pulse Type was not set to Step/Dir.
							1	1								Configuration tried to simultaneously enable the Quadrature Encoder and Diagnostic Feedback
					1			0								Configuration tried to request a <i>Home to Marker Pulse</i> setting without enabling the Quadrature Encoder.
					1			1	0					0		Configuration tried to request a <i>Home to Marker Pulse</i> setting without enabling the Home Input or Backplane Proximity bit. (Marker pulse homing requires one of the two for home proximity sensing.)
					1			1	1						1	Configuration tried to request a <i>Home to Marker Pulse</i> setting while enabling both the Home Input and the Backplane Proximity bit. (Marker pulse homing requires one and only one of the two for home proximity sensing.)

Table R7.2 Configuration Word 0 Invalid Combinations

Configuration Word 1

Word 1 configures Axis 1, Word 9 configures Axis 2

If these bits are set as shown....																...a Configuration Error will result because:
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	1	1	1	1	1	1	1	1	1	1						ANY of the reserved bits are set

Table R7.3 Configuration Bits LSW Errors

*Starting Speed Parameter*The Starting Speed parameter must not be outside its range of 1 to 1,000,000. A value of zero *will* cause an error.*Homing Timeout Parameter*

This parameter must be in the range of 0 to 300 decimal.

Reserved Words

The five reserved words in the configuration programming blocks for each axis must always be zero.

Notes

REFERENCE 8: COMMAND MODE OUTPUT DATA FORMAT

The custom macros supplied by IDEC simplify the configuration and programming of the IANF2(E) units. The data in this reference is presented as background information to aid in troubleshooting.

Data Format

Thirty-two bit values are transmitted most significant word first (big endian). This is the data format specified by the Modbus specification and used by IDEC controllers. Table R7.1 below shows the format of the data as it is stored and transmitted.

Value	32 bit Integer Format	
	First Word	Second Word
12	0x0000	0x000C
-12	0xFFFF	0xFF4
1,234,567	0x0012	0xD687
-7,654,321	0xFF8B	0x344F

Table R8.1 Position Data Format Examples

Command Bits Must Transition

Note ▶ Commands are only accepted when the command bit makes a 0→1 transition. The easiest way to do this is to write a value of zero into the Command Word 0 before writing the next command.

This condition also applies when switching from Configuration Mode to Command Mode. If a bit is set in Configuration Word 0 while in Configuration Mode and you switch to Command Mode with the same bit set, the command will not occur because the bit must transition between writes to the unit.

The command bits are split between two 16 bit words, Command Word 0 and Command Word 1. Only one command bit can make a 0→1 transition at a time.

Single Axis Commands

The following table shows the format of the output data when writing command data to the IANF2(E) that affects a single axis.

Offset from 1st Write Register Address for Axis ⁽¹⁾	Command Mode Input Data
0	Command Word 0
1	Command Word 1
2	Command Parameters: Word meaning depends on the command sent to the axis of the IANF2(E)
3	
4	
5	
6	
7	
8	
9	

(1) When using the User-defined Macros, the first Modbus Write Register address is mapped into the range of registers D49120 to D49220. See table T5.1, *Reserved Memory Addresses*, on page 59 for a list of the addresses for each axis.

Figure R8.1 Command Data Format

Command Word 0

The Command Word 0 is used to control the motion of the axis. Consequently, only one bit in the word can be on at a time. AMCI strongly suggests that you use MOV instructions when setting this word instead of bit instructions to set and reset the individual bits.

Command Word Value (hex)	Command Word Value (decimal)	Bit # set	Description
0x0000	0	None	This word must equal zero when performing any Interpolated Move.
0x0001	1	0	Perform an Absolute Move
0x0002	2	1	Perform an Relative Move
0x0004	4	2	Perform a Hold Move operation
0x0008	8	3	Perform a Resume Move operation
0x0010	16	4	Perform an Immediate Stop operation
0x0020	32	5	Perform a Home CW operation
0x0040	64	6	Perform a Home CCW operation
0x0080	128	7	Perform one of the following based on Command Parameter data: 1) Jog CW 2) Registration Move CW 3) Encoder Follower CW
0x0100	256	8	Perform one of the following based on Command Parameter data: 1) Jog CCW 2) Registration Move CCW 3) Encoder Follower CCW
0x0200	512	9	Preset Position command. This command presets the motor position. This command can also be used to reset the <i>Move Complete</i> status bit.
0x0400	1024	10	Preset Encoder Position command
0x0800	2048	11	Reset Command Errors
0x1000	4096	12	Run Blend Move CW
0x2000	8192	13	Run Blend Move CCW
0x4000	16384	14	Set Minimum Registration Move Distance

Table R8.2 Command Bits MSW

The remaining bit of this word is bit 15.

- ▶ **Bit 15: MODE CONTROL Bit:** A '0' places the axis of the IANF2(E) in Command Mode. This bit must be '0' when running any of the above commands. A '1' places the axis of the IANF2(E) in Configuration Mode.

Command Word 1

To Make These Settings...		...Set These Bits To															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Acceleration Type	Constant															0	0
	Constant															1	0
	Triangular S															0	1
	Trapezoidal S															1	1
No_Fault Output when Set_State = "1"	Off														0		
	On														1		
On_Line Output† when Set_State = "1"	Off													0			
	On													1			
Move_Complete when Set_State = "1"	Off											0					
	On											1					
Set_State	Off											0					
	On											1					
Backplane Home Proximity	Not Near Home										0						
	Near Home										1						
Program Blend Profile Transmit	Accept New Block											0→1					
	Acknowledge OK'd											1→0					
Encoder Follower Mode	Disabled									0							
	Enabled									1							
Run any command in Command Bits MSW			0	0	0	0	0	0									
Run a Relative Interpolated Move								0									
Run an Absolute Interpolated Move								1									
Run a Linear Interpolated Move			0	0	0	0	1	0/1									
Run a Via Point Circular Interpolated Move			0	0	0	1	0	0/1									
Run a CW Center Point Circular Int. Move			0	0	1	0	0	0/1									
Run a CCW Center Point Circular Int. Move			0	1	0	0	0	0/1									
Run a CW Radius Circular Int. Move			1	0	1	0	0	0/1									
Run a CCW Radius Circular Int. Move			1	1	0	0	0	0/1									
Program Blend Profile	Disabled	0															
	Enabled	1															

† Axis 1 only. This is the General Purpose Output for Axis 2 on an IANF2(E).

Table R8.3 Command Word 1

Single Axis Command Blocks

The following section lists the output data format for the fourteen different commands.

Absolute Move

Offset from 1st Write Register Address for Axis	Function	Units	Range
0	<i>Command Word 0</i>		0x0001
1	<i>Command Word 1</i>		See pg. 93
2	Target Position MSW	Steps	±1,073,741,823 (0xC000:0001 to 0x3FFF:FFFF)
3	Target Position LSW		
4	Programmed Speed MSW	Steps/Second	Configured Starting Speed to 1,000,000 (0x000F:4240)
5	Programmed Speed LSW		
6	Acceleration	Steps/ms/sec	1 to 2000
7	Deceleration	Steps/ms/sec	1 to 2000
8	<i>Reserved</i>		0x0000
9	<i>Reserved</i>		0x0000

Table R8.4 Absolute Move Command Block

Relative Move

Offset from 1st Write Register Address for Axis	Function	Units	Range
0	<i>Command Word 0</i>		0x0002
1	<i>Command Word 1</i>		See pg. 93
2	Relative Position MSW	Steps	±1,073,741,823 (0xC000:0001 to 0x3FFF:FFFF)
3	Relative Position LSW		
4	Programmed Speed MSW	Steps/Second	Configured Starting Speed to 1,000,000 (0x000F:4240)
5	Programmed Speed LSW		
6	Acceleration	Steps/ms/sec	1 to 2000
7	Deceleration	Steps/ms/sec	1 to 2000
8	<i>Reserved</i>		0x0000
9	<i>Reserved</i>		0x0000

Table R8.5 Relative Move Command Block

Hold Move

Offset from 1st Write Register Address for Axis	Function	Units	Range
0	<i>Command Word 0</i>		0x0004
1	<i>Command Word 1</i>		See pg. 93
2	Unused		0 (See Note 1 below)
3	Unused		0 (See Note 1 below)
4	Unused		0 (See Note 1 below)
5	Unused		0 (See Note 1 below)
6	Unused		0 (See Note 1 below)
7	Unused		0 (See Note 1 below)
8	Unused		0 (See Note 1 below)
9	Unused		0 (See Note 1 below)

(1) Unused words are ignored by the IANF2(E) and can be any value, including parameter values from the previous command. These registers are typically reset to zero by the IDEC supplied User-defined macros.

Table R8.6 Hold Move Command Block

Resume Move

Offset from 1st Write Register Address for Axis	Function	Units	Range
0	<i>Command Word 0</i>		0x0008
1	<i>Command Word 1</i>		See pg. 93
2	Unused		0 (See Note 1 below)
3	Unused		0 (See Note 1 below)
4	Programmed Speed MSW	Steps/Second	Configured Starting Speed to 1,000,000 (0x000F:4240)
5	Programmed Speed LSW		
6	Acceleration	Steps/ms/sec	1 to 2000
7	Deceleration	Steps/ms/sec	1 to 2000
8	Unused		0 (See Note 1 below)
9	Unused		0 (See Note 1 below)

(1) Unused words are ignored by the IANF2(E) and can be any value, including parameter values from the previous command. These registers are typically reset to zero by the IDEC supplied User-defined macros.

Table R8.7 Resume Move Command Block

Immediate Stop

Offset from 1st Write Register Address for Axis	Function	Units	Range
0	<i>Command Word 0</i>		0x0010
1	<i>Command Word 1</i>		See pg. 93 and Note 1 below
2	Unused		0 (See Note 2 below)
3	Unused		0 (See Note 2 below)
4	Unused		0 (See Note 2 below)
5	Unused		0 (See Note 2 below)
6	Unused		0 (See Note 2 below)
7	Unused		0 (See Note 2 below)
8	Unused		0 (See Note 2 below)
9	Unused		0 (See Note 2 below)

- (1) When running an Interpolated Move, the IANF2(E) requires an Immediate Stop command on channel 1. An Immediate Stop instruction on channel 1 stops the move and the motor position values on both axes becomes invalid. An Immediate Stop instruction on channel 2 will not stop the move but the IANF2(E) will issue a Command Error message on channel 1.
- (2) Unused words are ignored by the IANF2(E) and can be any value, including parameter values from the previous command. These registers are typically reset to zero by the IDEC supplied User-defined macros.

Table R8.8 Immediate Stop Command Block

Home CW

Offset from 1st Write Register Address for Axis	Function	Units	Range
0	<i>Command Word 0</i>		0x0020
1	<i>Command Word 1</i>		See pg. 93 and Note 1 below
2	<i>Reserved</i>		<i>Must equal 0x0000 for compatibility with future releases.</i>
3	<i>Reserved</i>		<i>Must equal 0x0000 for compatibility with future releases.</i>
4	Programmed Speed MSW	Steps/Second	Configured Starting Speed to 1,000,000 (0x000F:4240)
5	Programmed Speed LSW		
6	Acceleration	Steps/ms/sec	1 to 2000
7	Deceleration	Steps/ms/sec	1 to 2000
8	<i>Reserved</i>		<i>Must equal 0x0000 for compatibility with future releases.</i>
9	<i>Reserved</i>		<i>Must equal 0x0000 for compatibility with future releases.</i>

- (1) Only Constant Acceleration is available when homing an IANF2(E) axis. Bit 0 of the Command Word 1 must equal zero.

Table R8.9 Home CW Command Block

Home CCW

Offset from 1st Write Register Address for Axis	Function	Units	Range
0	<i>Command Word 0</i>		0x0040
1	<i>Command Word 1</i>		See pg. 93 and note 1 below
2	<i>Reserved</i>		<i>Must equal 0x0000 for compatibility with future releases.</i>
3	<i>Reserved</i>		<i>Must equal 0x0000 for compatibility with future releases.</i>
4	Programmed Speed MSW	Steps/Second	Configured Starting Speed to 1,000,000 (0x000F:4240)
5	Programmed Speed LSW		
6	Acceleration	Steps/ms/sec	1 to 2000
7	Deceleration	Steps/ms/sec	1 to 2000
8	<i>Reserved</i>		<i>Must equal 0x0000 for compatibility with future releases.</i>
9	<i>Reserved</i>		<i>Must equal 0x0000 for compatibility with future releases.</i>

(1) Only Constant Acceleration is available when homing an IANF2(E) axis. Bit 0 of the Command Word 1 must equal zero.

Table R8.10 Home CW Command Block

Jog CW

Offset from 1st Write Register Address for Axis	Function	Units	Range
0	<i>Command Word 0</i>		0x0080
1	<i>Command Word 1</i>		See pg. 93
2	<i>Reserved</i>		<i>Must equal 0x0000 for Jog Moves.</i>
3	<i>Reserved</i>		<i>Must equal 0x0000 for Jog Moves.</i>
4	Programmed Speed MSW	Steps/Second	Configured Starting Speed to 1,000,000 (0x000F:4240)
5	Programmed Speed LSW		
6	Acceleration	Steps/ms/sec	1 to 2000
7	Deceleration	Steps/ms/sec	1 to 2000
8	<i>Reserved</i>		<i>Must equal 0x0000 for compatibility with future releases.</i>
9	<i>Reserved</i>		<i>Must equal 0x0000 for compatibility with future releases.</i>

Table R8.11 Jog CW Command Block

Registration Move CW

Offset from 1st Write Register Address for Axis	Function	Units	Range
0	<i>Command Word 0</i>		0x0080
1	<i>Command Word 1</i>		See pg. 93
2	Position Offset MSW	Steps	1 to 1,073,741,823 1 to 0x3FFF:FFFF
3	Position Offset LSW		
4	Programmed Speed MSW	Steps/Second	Configured Starting Speed to 1,000,000 (0x000F:4240)
5	Programmed Speed LSW		
6	Acceleration	Steps/ms/sec	1 to 2000
7	Deceleration	Steps/ms/sec	1 to 2000
8	<i>Reserved</i>		<i>Must equal 0x0000 for compatibility with future releases.</i>
9	<i>Reserved</i>		<i>Must equal 0x0000 for compatibility with future releases.</i>

Table R8.12 Registration Move CW Command Block

Encoder Follower CW

Offset from 1st Write Register Address for Axis	Function	Units	Range
0	<i>Command Word 0</i>		0x0080
1	<i>Command Word 1</i>		0x0100
2	Encoder Follower Multiplier		1 to 32,767
3	Encoder Follower Divisor		1 to 32,767
4	<i>Reserved</i>		<i>Must equal 0x0000 for compatibility with future releases.</i>
5	<i>Reserved</i>		<i>Must equal 0x0000 for compatibility with future releases.</i>
6	Acceleration	Steps/ms/sec	1 to 2000
7	Deceleration	Steps/ms/sec	1 to 2000
8	<i>Reserved</i>		<i>Must equal 0x0000 for compatibility with future releases.</i>
9	<i>Reserved</i>		<i>Must equal 0x0000 for compatibility with future releases.</i>

Table R8.13 Encoder Follower CW Command Block

Jog CCW

Offset from 1st Write Register Address for Axis	Function	Units	Range
0	<i>Command Word 0</i>		0x0100
1	<i>Command Word 1</i>		See pg. 93
2	<i>Reserved</i>		<i>Must equal 0x0000 for Jog Moves.</i>
3	<i>Reserved</i>		<i>Must equal 0x0000 for Jog Moves.</i>
4	Programmed Speed MSW	Steps/Second	Configured Starting Speed to 1,000,000 (0x000F:4240)
5	Programmed Speed LSW		
6	Acceleration	Steps/ms/sec	1 to 2000
7	Deceleration	Steps/ms/sec	1 to 2000
8	<i>Reserved</i>		<i>Must equal 0x0000 for compatibility with future releases.</i>
9	<i>Reserved</i>		<i>Must equal 0x0000 for compatibility with future releases.</i>

Table R8.14 Jog CCW Command Block

Registration Move CCW

Offset from 1st Write Register Address for Axis	Function	Units	Range
0	<i>Command Word 0</i>		0x0100
1	<i>Command Word 1</i>		See pg. 93
2	Position Offset MSW	Steps	1 to 1,073,741,823 1 to 0x3FFF:FFFF
3	Position Offset LSW		
4	Programmed Speed MSW	Steps/Second	Configured Starting Speed to 1,000,000 (0x000F:4240)
5	Programmed Speed LSW		
6	Acceleration	Steps/ms/sec	1 to 2000
7	Deceleration	Steps/ms/sec	1 to 2000
8	<i>Reserved</i>		<i>Must equal 0x0000 for compatibility with future releases.</i>
9	<i>Reserved</i>		<i>Must equal 0x0000 for compatibility with future releases.</i>

Table R8.15 Registration Move CCW Command Block

Encoder Follower CCW

Offset from 1st Write Register Address for Axis	Function	Units	Range
0	<i>Command Word 0</i>		0x0100
1	<i>Command Word 1</i>		0x0100
2	Encoder Follower Multiplier		1 to 32,767
3	Encoder Follower Divisor		1 to 32,767
4	<i>Reserved</i>		<i>Must equal 0x0000 for compatibility with future releases.</i>
5	<i>Reserved</i>		<i>Must equal 0x0000 for compatibility with future releases.</i>
6	Acceleration	Steps/ms/sec	1 to 2000
7	Deceleration	Steps/ms/sec	1 to 2000
8	<i>Reserved</i>		<i>Must equal 0x0000 for compatibility with future releases.</i>
9	<i>Reserved</i>		<i>Must equal 0x0000 for compatibility with future releases.</i>

Table R8.16 Encoder Follower CCW Command Block

Preset Position

Offset from 1st Write Register Address for Axis	Function	Units	Range
0	<i>Command Word 0</i>		0x0200 ⁽¹⁾
1	<i>Command Word 1</i>		See pg. 93
2	New Motor Position MSW	Steps	±1,073,741,823 (0xC000:0001 to 0x3FFF:FFFF)
3	New Motor Position LSW		
4	Unused		0 (See Note 2 below)
5	Unused		0 (See Note 2 below)
6	Unused		0 (See Note 2 below)
7	Unused		0 (See Note 2 below)
8	Unused		0 (See Note 2 below)
9	Unused		0 (See Note 2 below)

(1) Presetting the position will also reset the Move Complete status bit.

(2) Unused words are ignored by the IANF2(E) and can be any value, including parameter values from the previous command. These registers are typically reset to zero by the IDEC supplied User-defined macros.

Table R8.17 Preset Position Command Block

Preset Encoder Position

Offset from 1st Write Register Address for Axis	Function	Units	Range
0	<i>Command Word 0</i>		0x0400
1	<i>Command Word 1</i>		See pg. 93
2	New Encoder Position MSW	Steps	±1,073,741,823 (0xC000:0001 to 0x3FFF:FFFF)
3	New Encoder Position LSW		
4	Unused		0 (See Note 1 below)
5	Unused		0 (See Note 1 below)
6	Unused		0 (See Note 1 below)
7	Unused		0 (See Note 1 below)
8	Unused		0 (See Note 1 below)
9	Unused		0 (See Note 1 below)

- (1) Unused words are ignored by the IANF2(E) and can be any value, including parameter values from the previous command. These registers are typically reset to zero by the IDEC supplied User-defined macros.

Table R8.18 Preset Encoder Position Command Block

Reset Errors

Offset from 1st Write Register Address for Axis	Function	Units	Range
0	<i>Command Word 0</i>		0x0800
1	<i>Command Word 1</i>		See pg. 93
2	Unused		0 (See Note 1 below)
3	Unused		0 (See Note 1 below)
4	Unused		0 (See Note 1 below)
5	Unused		0 (See Note 1 below)
6	Unused		0 (See Note 1 below)
7	Unused		0 (See Note 1 below)
8	Unused		0 (See Note 1 below)
9	Unused		0 (See Note 1 below)

- (1) Unused words are ignored by the IANF2(E) and can be any value, including parameter values from the previous command. These registers are typically reset to zero by the IDEC supplied User-defined macros.

Table R8.19 Reset Errors Command Block

Blend Move CW

Offset from 1st Write Register Address for Axis	Function	Units	Range
0	<i>Command Word 0</i>		0x1000
1	<i>Command Word 1</i>		See pg. 93
2	Unused		0 (See Note 1 below)
3	Unused		0 (See Note 1 below)
4	Unused		0 (See Note 1 below)
5	Unused		0 (See Note 1 below)
6	Unused		0 (See Note 1 below)
7	Unused		0 (See Note 1 below)
8	Unused		0 (See Note 1 below)
9	Unused		0 (See Note 1 below)

- (1) Unused words are ignored by the IANF2(E) and can be any value, including parameter values from the previous command. These registers are typically reset to zero by the IDEC supplied User-defined macros.

Table R8.20 Blend Move CW Command Block

Blend Move CCW

Offset from 1st Write Register Address for Axis	Function	Units	Range
0	<i>Command Word 0</i>		0x2000
1	<i>Command Word 1</i>		See pg. 93
2	Unused		0 (See Note 1 below)
3	Unused		0 (See Note 1 below)
4	Unused		0 (See Note 1 below)
5	Unused		0 (See Note 1 below)
6	Unused		0 (See Note 1 below)
7	Unused		0 (See Note 1 below)
8	Unused		0 (See Note 1 below)
9	Unused		0 (See Note 1 below)

(1) Unused words are ignored by the IANF2(E) and can be any value, including parameter values from the previous command. These registers are typically reset to zero by the IDEC supplied User-defined macros.

Table R8.21 Blend Move CCW Command Block

Set Minimum Registration Move Distance

Offset from 1st Write Register Address for Axis	Function	Units	Range
0	<i>Command Word 0</i>		0x4000
1	<i>Command Word 1</i>		See pg. 93
2	Minimum Registration Distance MSW	Steps	0 to 1,073,741,823 (0x0000:0000 to 0x3FFF:FFFF)
3	Minimum Registration Distance MSW		
4	Unused		0 (See Note 1 below)
5	Unused		0 (See Note 1 below)
6	Unused		0 (See Note 1 below)
7	Unused		0 (See Note 1 below)
8	Unused		0 (See Note 1 below)
9	Unused		0 (See Note 1 below)

(1) Unused words are ignored by the IANF2(E) and can be any value, including parameter values from the previous command. These registers are typically reset to zero by the IDEC supplied User-defined macros.

Table R8.22 Set Minimum Registration Move Distance Command Block

Interpolated Move Command Blocks

Linear Interpolated Move

Offset from 1st Write Register Address for Card	Function	Units	Range
0	<i>Command Word 0</i>		0x0000
1	<i>Command Word 1</i>		See pg. 93 and Note (1) below.
2	Interpolated Speed MSW	Steps/second	Starting Speed to 1,000,000 (0x000F:4240)
3	Interpolated Speed MSW		
4	Interpolated Acceleration Value	Steps/ms/second	1 to 2000
5	Interpolated Deceleration Value	Steps/ms/second	1 to 2000
6	EndPointX MSW (axis 1)	Steps	±1,073,741,823 (0xC000:0001 to 0x3FFF:FFFF)
7	EndPointX LSW (axis 1)		
8	<i>Reserved</i>		<i>Must equal 0x0000.</i>
9	<i>Reserved</i>		<i>Must equal 0x0000.</i>
10	<i>Reserved</i>		<i>Must equal 0x0000.</i>
11	<i>Reserved</i>		<i>Must equal 0x0000.</i>
12	EndPointY MSW (axis 2)	Steps	±1,073,741,823 (0xC000:0001 to 0x3FFF:FFFF)
13	EndPointY LSW (axis 2)		
14	<i>Reserved</i>		<i>Must equal 0x0000.</i>
15	<i>Reserved</i>		<i>Must equal 0x0000.</i>
16	<i>Reserved</i>		<i>Must equal 0x0000.</i>
17	<i>Reserved</i>		<i>Must equal 0x0000.</i>
18	<i>Reserved</i>		<i>Must equal 0x0000.</i>
19	<i>Reserved</i>		<i>Must equal 0x0000.</i>

(1) The *Command Word 1*, fully described in page 93, contains the command bits as well as the Acceleration Type parameter.

- For a Relative Linear Interpolated Move, the upper byte of the Command Bits LSW must equal 16#04. (Bit 10 set)
- For an Absolute Linear Interpolated Move, the upper byte of the Command Bits LSW must equal 16#06. (Bits 10 and 9 set)
- For Constant Acceleration, the lower two bits of the Command Bits LSW can equal 2#00 or 2#10. (Both bits reset or bit 1 set)
- For Triangular S-Curve Acceleration, the lower two bits of the Command Bits LSW must equal 2#01. (Bit 0 set)
- For Trapezoidal S-Curve Acceleration, the lower two bits of the Command Bits LSW must equal 2#11. (Bits 1 and 0 set)

Table R8.23 Linear Interpolated Move Command Block

Via Point Circular Interpolated Move

Offset from 1st Write Register Address for Card	Function	Units	Range
0	<i>Command Word 0</i>		0x0000
1	<i>Command Word 1</i>		See pg. 93 and Note (1) below.
2	Interpolated Speed MSW	Steps/second	Starting Speed to 1,000,000 (0x000F:4240)
3	Interpolated Speed LSW		
4	Interpolated Acceleration Value	Steps/ms/second	1 to 2000
5	Interpolated Deceleration Value	Steps/ms/second	1 to 2000
6	EndPointX MSW (axis 1)	Steps	±1,073,741,823 (0xC000:0001 to 0x3FFF:FFFF)
7	EndPointX LSW (axis 1)		
8	<i>Reserved</i>		<i>Must equal 0x0000.</i>
9	<i>Reserved</i>		<i>Must equal 0x0000.</i>
10	<i>Reserved</i>		<i>Must equal 0x0000.</i>
11	<i>Reserved</i>		<i>Must equal 0x0000.</i>
12	EndPointY MSW (axis 2)	Steps	±1,073,741,823 (0xC000:0001 to 0x3FFF:FFFF)
13	EndPointY LSW (axis 2)		
14	ViaPointX MSW (axis 1)	Steps	±1,073,741,823 (0xC000:0001 to 0x3FFF:FFFF)
15	ViaPointX LSW (axis 1)		
16	ViaPointY MSW (axis 2)	Steps	±1,073,741,823 (0xC000:0001 to 0x3FFF:FFFF)
17	ViaPointY LSW (axis 2)		
18	<i>Reserved</i>		<i>Must equal 0x0000.</i>
19	<i>Reserved</i>		<i>Must equal 0x0000.</i>

(1) The *Command Word 1*, fully described in page 93, contains the command bits as well as the Acceleration Type parameter.

- For a Relative Via Point Circular Interpolated Move, the upper byte of the Command Bits LSW must equal 16#08. (Bit 11 set)
- For an Absolute Via Point Circular Interpolated Move, the upper byte of the Command Bits LSW must equal 16#0A. (Bits 11 and 9 set)
- For Constant Acceleration, the lower two bits of the Command Bits LSW can equal 2#00 or 2#10. (Both bits reset or bit 1 set)
- For Triangular S-Curve Acceleration, the lower two bits of the Command Bits LSW must equal 2#01. (Bit 0 set)
- For Trapezoidal S-Curve Acceleration, the lower two bits of the Command Bits LSW must equal 2#11. (Bits 1 and 0 set)

Table R8.24 Via Point Circular Interpolated Move Command Block

Center Point Circular Interpolated Move

Offset from 1st Write Register Address for Card	Function	Units	Range
0	<i>Command Word 0</i>		0x0000
1	<i>Command Word 1</i>		See pg. 93 and Note (1) below.
2	Interpolated Speed MSW	Steps/second	Starting Speed to 1,000,000 (0x000F:4240)
3	Interpolated Speed LSW		
4	Interpolated Acceleration Value	Steps/ms/second	1 to 2000
5	Interpolated Deceleration Value	Steps/ms/second	1 to 2000
6	EndPointX MSW (axis 1)	Steps	±1,073,741,823 (0xC000:0001 to 0x3FFF:FFFF)
7	EndPointX LSW (axis 1)		
8	<i>Reserved</i>		<i>Must equal 0x0000.</i>
9	<i>Reserved</i>		<i>Must equal 0x0000.</i>
10	<i>Reserved</i>		<i>Must equal 0x0000.</i>
11	<i>Reserved</i>		<i>Must equal 0x0000.</i>
12	EndPointY MSW (axis 2)	Steps	±1,073,741,823 (0xC000:0001 to 0x3FFF:FFFF)
13	EndPointY LSW (axis 2)		
14	CenterPointX MSW (axis 1)	Steps	±1,073,741,823 (0xC000:0001 to 0x3FFF:FFFF)
15	CenterPointX LSW (axis 1)		
16	CenterPointY MSW (axis 2)	Steps	±1,073,741,823 (0xC000:0001 to 0x3FFF:FFFF)
17	CenterPointY LSW (axis 2)		
18	<i>Reserved</i>		<i>Must equal 0x0000.</i>
19	<i>Reserved</i>		<i>Must equal 0x0000.</i>

(1) The *Command Word 1*, fully described in page 93, contains the command bits as well as the Acceleration Type parameter.

- For a Relative Center Point Circular Interpolated Move that moves in a clockwise direction, the upper byte of the Command Word LSW must equal 16#10. (Bit 12 set)
- For a Relative Center Point Circular Interpolated Move that moves in a counter-clockwise direction, the upper byte of the word must equal 16#20. (Bit 13 set)
- For an Absolute Center Point Circular Interpolated Move that moves in a clockwise direction, the upper byte of the word must equal 16#12. (Bits 12 and 9 set)
- For an Absolute Center Point Circular Interpolated Move that moves in a counter-clockwise direction, the upper byte of the word must equal 16#22. (Bits 13 and 9 set)
- For Constant Acceleration, the lower two bits of the Command Bits LSW can equal 2#00 or 2#10. (Both bits reset or bit 1 set)
- For Triangular S-Curve Acceleration, the lower two bits of the Command Bits LSW must equal 2#01. (Bit 0 set)
- For Trapezoidal S-Curve Acceleration, the lower two bits of the Command Bits LSW must equal 2#11. (Bits 1 and 0 set)

Table R8.25 Center Point Circular Interpolated Move Command Block

Radius Circular Interpolated Move

Offset from 1st Write Register Address for Card	Function	Units	Range
0	<i>Command Word 0</i>		0x0000
1	<i>Command Word 1</i>		See pg. 93 and Note (1) below.
2	Interpolated Speed MSW	Steps/second	Starting Speed to 1,000,000 (0x000F:4240)
3	Interpolated Speed LSW		
4	Interpolated Acceleration Value	Steps/ms/second	1 to 2000
5	Interpolated Deceleration Value	Steps/ms/second	1 to 2000
6	EndPointX MSW (axis 1)	Steps	±1,073,741,823 (0xC000:0001 to 0x3FFF:FFFF)
7	EndPointX LSW (axis 1)		
8	<i>Reserved</i>		<i>Must equal 0x0000.</i>
9	<i>Reserved</i>		<i>Must equal 0x0000.</i>
10	<i>Reserved</i>		<i>Must equal 0x0000.</i>
11	<i>Reserved</i>		<i>Must equal 0x0000.</i>
12	EndPointY MSW (axis 2)	Steps	±1,073,741,823 (0xC000:0001 to 0x3FFF:FFFF)
13	EndPointY LSW (axis 2)		
14	Radius MSW	Steps	±1,073,741,823 (0xC000:0001 to 0x3FFF:FFFF) (See Note 2)
15	Radius LSW		
16	<i>Reserved</i>		<i>Must equal 0x0000.</i>
17	<i>Reserved</i>		<i>Must equal 0x0000.</i>
18	<i>Reserved</i>		<i>Must equal 0x0000.</i>
19	<i>Reserved</i>		<i>Must equal 0x0000.</i>

(1) The *Command Word 1*, fully described in page 93, contains the command bits as well as the Acceleration Type parameter.

- For a Relative, Radius Circular Interpolated Move that travels in a clockwise direction, the upper byte of the word must equal 16#50. (Bits 14 and 12 set)
- For a Relative, Radius Circular Interpolated Move that travels in a counter-clockwise direction, the upper byte of the word must equal 16#60. (Bits 14 and 13 set)
- For an Absolute, Radius Circular Interpolated Move that travels in a clockwise direction, the upper byte of the word must equal 16#52. (Bits 14, 12, and 9 set)
- For an Absolute, Radius Circular Interpolated Move that travels in a counter-clockwise direction, the upper byte of the word must equal 16#62. (Bits 14, 13, and 9 set)
- For Constant Acceleration, the lower two bits of the Command Bits LSW can equal 2#00 or 2#10. (Both bits reset or bit 1 set)
- For Triangular S-Curve Acceleration, the lower two bits of the Command Bits LSW must equal 2#01. (Bit 0 set)
- For Trapezoidal S-Curve Acceleration, the lower two bits of the Command Bits LSW must equal 2#11. (Bits 1 and 0 set)

(2) The Radius method defines two circles on the plane. (See *Radius Method* on page 33 for a full description of this method.) If the Radius value is positive, the move will travel the shortest arc in the specified direction. If the Radius value is negative, the move will travel the longest arc in the specified direction.

Table R8.26 Radius Circular Interpolated Move Command Block

REFERENCE 9: BLEND MOVES

Introduction

The IANF2(E) contains functionality that is not programmable through custom macros at this time. This reference section describes this additional functionality.

Blend Moves

All of the moves explained so far must be run individually to their completion or must be stopped before another move can begin. The IANF2(E) also gives you the ability to pre-assemble more complex profiles from a series of relative moves. Each relative move defines a *segment of the Blend Move*, and each move consists of two to sixteen segments. A single Blend Move command will then run all of the relative moves sequentially without stopping the shaft between segments. A Blend Move can be run in either direction, and the direction is set when the command is issued. The direction of motion cannot be reversed within a Blend Move. Blend Moves are programmed through a hand shaking protocol that uses the input and output registers assigned to the unit. The protocol is fully described in the *Programming Blend Move Profiles* section on page 106.

The following restrictions apply when programming Blend Moves.

- 1) Each segment of the Blend Move must be written to the IANF2(E) before the move can be initiated.
- 2) Each segment is programmed as a relative move. Blend Moves cannot be programmed with absolute coordinates.
- 3) All segments run in the same direction. The sign of the target position is ignored and only the magnitude of the target position is used. The move's direction is controlled by the bit pattern used to start the move.
- 4) The Programmed Speed of each segment must be greater than or equal to the Starting Speed.
- 5) The Programmed Speed cannot be the same between segments.
- 6) For all segments except for the last one, the programmed position defines the end of the segment. For the last segment, the programmed position defines the end of the move.
- 7) Once you enter a segment, that segment's programmed acceleration/deceleration value is used to change the speed of the motor.
- 8) A Blend Segment Error is generated by the IANF2(E) if the segment's Programmed Speed is the same as the previous segment's, or if the number of steps in the segment is not enough to perform the programmed acceleration/deceleration within the segment. (For example, if the required number of steps to accelerate within the segment is 3,000, and the segment length is only 2,500, an error will result.)

The figure below shows a three segment Blend Move that is run twice. It is first run in the clockwise direction, and then in the counter-clockwise direction.

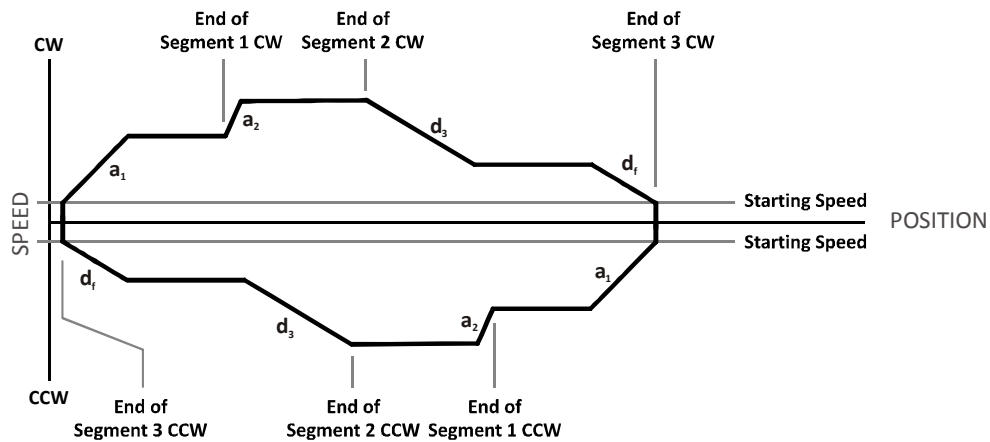


Figure R9.1 Blend Move

- Note** ▶
- 1) The final deceleration value (D_f) is programmed separately from the final segment. The Acceleration/Deceleration value programmed as part of the last segment is only used to accelerate or decelerate at the beginning of the segment.
 - 2) The home position of the machine need not be defined to use a Blend Move. Because the Blend Move is based on Relative Moves, it can be run from any location.
 - 3) The Blend Move is stored in the internal memory of the IANF2(E) and can be run multiple times once it is written to the unit. The Blend Move data stays in memory until power is removed, the unit is sent new Configuration Data, or a new Blend or Dwell Move is written to the unit.
 - 4) There are two control bits used to specify which direction the Blend Move is run in. This gives you the ability to run the Blend Move in either direction.

Controlled Stop Conditions

- The move completes without error.

Immediate Stop Conditions

- The Immediate_Stop bit makes a 0→1 transition in the Network Output Data.
- A positive transition on an input configured as an E-Stop Input.
- A CW or CCW Limit Switch is reached.

Controlling a Blend Move In Progress

Blend Moves cannot be brought into a Hold State. The External Input is ignored during a Blend Move and issuing a Hold command will only result in the Command_Error bit being set. The move profile will continue until it is complete, at which time the Command_Error bit will be reset.

Programming Blend Move Profiles

A Blend Move Profile must be programmed into the IANF2(E) before a Blend Move can be executed. Multiple writes to the IANF2(E) are required to program a single Blend Move Profile. The IANF2(E) uses a simple handshaking protocol to control this transfer of data. Two bits in the Output Network Data and two bits in the Input Network Data are used in the transfer of data blocks.

The two bits in the Output Network Data were introduced in the *Command Word 1* section on page 93.

- **Program_Blend_Profile** (Bit 15 of Command Word 1) Must make a 0→1 transition to start the Blend Move Profile programming cycle and must remain a "1" while programming the Blend Move Profile.
- **Program_Blend_Profile_Transmit** (Bit 7 of Command Word 1) Make a 0→1 transition on this bit to tell the IANF2(E) that new Blend Move Profile data is available. Make a 1→0 transition on this bit after the IANF2(E) sets its *Program Blend Profile Acknowledge* bit in its Input Network Data data to complete the transfer of the block of data.

The two bits in the Input Network Data are introduced here. The remaining values in the Input Network Data are introduced in the following section.

- **Waiting_For_Profile_Segment** (Bit 9 of Status Word 1) This bit is used by the IANF2(E) to tell the host controller when it has accepted a new data block, and when it is ready to receive the next data block.
- **Profile_Error** (Bit 9 of Status Word 0) This bit is used by the IANF2(E) to tell the host controller when there was an error in the last Program Blend Profile data block sent to the module.

The following logical steps are used to program a Blend Move Profile.

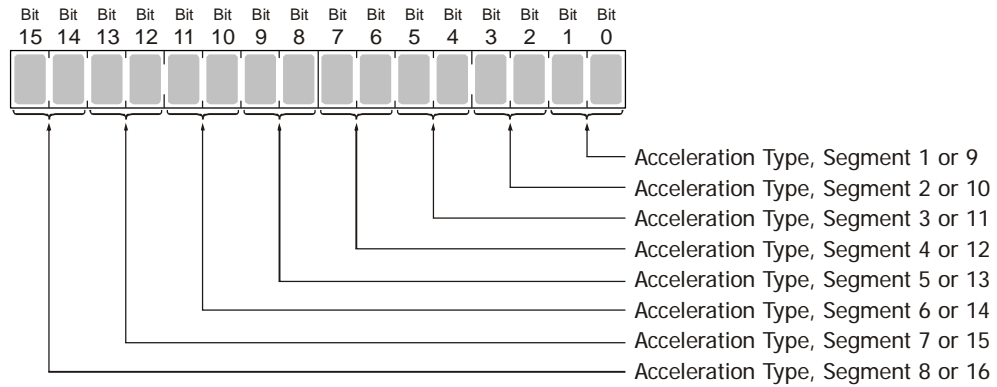
- 1) Your host controller's program writes the *Initial Blend Profile Programming Block* to the IANF2(E) with the *Program Blend Profile* and *Program Blend Profile Transmit* bits both set to 1.
- 2) The IANF2(E) responds by storing the information in the data block, setting the *Invalid Blend Profile* bit if there is an error in the data, and setting the *Program Blend Profile Acknowledge* bit.
- 3) Your host controller's program must detect when the *Program Blend Profile Acknowledge* bit is set. When this bit transitions, check the value of the *Invalid Blend Profile* bit and act appropriately.
- 4) Your host controller's program then needs to reset the *Program Blend Profile Transmit* bit.
- 5) The IANF2(E) responds by resetting its *Program Blend Profile Acknowledge* bit.
- 6) a: If you have additional data blocks that must be sent to the IANF2(E), repeat these instructions from step 1, sending the next *Secondary Profile Programming Block*.
b: If you have sent the last data block, then reset the *Program Blend Profile* bit to end the programming cycle.

Two different types of data blocks must be used when programming a Blend Move Profile. The first data block type programs the Number of Segments and the acceleration/deceleration types used between the segments. (Constant, Triangular S-Curve, Trapezoidal S-Curve) This must be the first data block written to the IANF2(E) when programming a Blend Move Profile. The remaining data block type is used to program the Segment Length, the Target Speed, and the actual acceleration/deceleration values. This block is used multiple times in the programming cycle, once for each segment in the profile.

Initial Blend Profile Programming Block

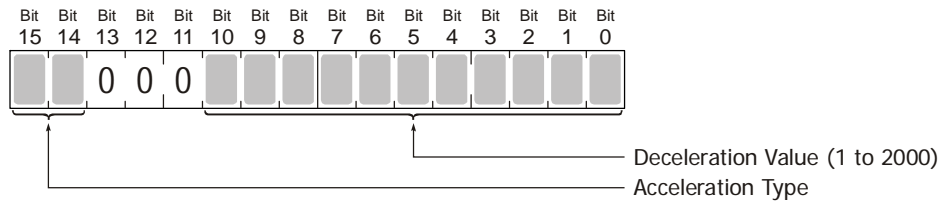
Offset from 1st Write Register Address for Axis	Function	Units	Range
0	<i>Command Word 0</i>		0x0000
1	<i>Command Word 1</i>		0x8080
2	Number of Segments		2 to 16
3	Acceleration Type MSW	Programs Acceleration Type for segments 9 - 16	
4	Acceleration Type LSW	Programs Acceleration Type for segments 1 - 8	
5	Deceleration Value and Type	Programs final deceleration value and type	
6	<i>Reserved</i>		<i>Must equal 0x0000.</i>
7	<i>Reserved</i>		<i>Must equal 0x0000.</i>
8	<i>Reserved</i>		<i>Must equal 0x0000.</i>
9	<i>Reserved</i>		<i>Must equal 0x0000.</i>

Table R9.1 Initial Blend Profile Programming Block



Bit Pair		Description
MSB	LSB	
0	0	Constant
0	1	Triangular S-Curve
1	0	Invalid Combination
1	1	Trapezoidal S-Curve

Figure R9.2 Acceleration Type MSW / LSW Word Format



Bit 15	Bit 14	Description
0	0	Constant
0	1	Triangular S-Curve
1	0	Invalid Combination
1	1	Trapezoidal S-Curve

Figure R9.3 Deceleration Value and Type Word Format

Secondary Profile Programming Block

Offset from 1st Write Register Address for Axis	Function	Units	Range
0	<i>Command Word 0</i>		0x0000
1	<i>Command Word 1</i>		0x8080
2	Segment Length MSW	Steps	1 to (1,073,741,823 - 1) ^{Notes 1 & 2} (0x0000:0001 to 0x3FFF:FFFE)
3	Segment Length LSW		
4	Target Speed MSW	Steps/second	Starting Speed to 1,000,000 (0x000F:4240)
5	Target Speed LSW		
6	Acceleration / Deceleration Value	Steps/ms/sec	1 to 2000
7	<i>Reserved</i>		<i>Must equal 0x0000.</i>
8	<i>Reserved</i>		<i>Must equal 0x0000.</i>
9	<i>Reserved</i>		<i>Must equal 0x0000.</i>

- (1) Combined length of all segments cannot exceed 1,073,741,823 counts.
- (2) The Segment Length must be long enough to achieve the Target Speed at the programmed Acceleration / Deceleration Value.

Table R9.2 Secondary Blend Profile Programming Block

Commanding a Blend Move

See *Blend Move CW* on page 99 and *Blend Move CCW* on page 100.

OPTIONAL TASK A: CONFIGURE YOUR NETWORK INTERFACES

Introduction

Before communicating with an IANF2(E) using a personal computer, you may need to adjust the network setting on the computer. This task section give you information on configuring your computer's network interfaces.

A.1 Firewall Settings

Firewalls are hardware devices or software that prevent unwanted network connections from occurring. Firewall software is present in Windows XP and above and it may prevent your computer for communicating with the IANF2(E). Configuring your firewall to allow communication with the IANF2(E) is beyond the scope of this manual.

It may be necessary to temporarily disable any firewall software while using the Ethernet Tool. This is typically done from the Windows Control Panel. You should re-enable the firewall once you have finished using the tool.

A.2 Disable All Unused Network Interfaces

Routing and default gateway setting on your computer can interfere with the proper operation of the Ethernet Tool. The software uses broadcast packets to locate devices on the network, and sometimes these packets are sent out through the default gateway instead of the interface attached to the IANF2(E). The easiest way to avoid this problem is to temporarily disable all network interfaces that are not attached to the stepper drive.

Note▶ This includes all wireless interfaces as well as all Bluetooth interfaces.

A.3 Configure Your Network Interface

Before you can communicate with the IANF2(E), your network interface must be on the same subnet as the module.

Note▶ The rest of this procedure assumes you are using the 192.168.1.xxx subnet. If you are not, you will have to adjust the given network addresses accordingly.

The easiest way to check the current settings for your NIC is with the 'ipconfig' command.

- ▶ For Vista and Windows 7, click on the [Start] button, and type "cmd" in the "Search programs and files" text box. Press [Enter] on the keyboard.
- ▶ For Windows 8 and 10, press the [Win+X] keys and select "Command Prompt" from the resulting popup. There is no need to run the command prompt as the administrator, so do not select "Command Prompt (Admin)".

A DOS like terminal will open. Type in 'ipconfig', press [Enter] on the keyboard and the computer will return the present Address, Subnet Mask, and Default Gateway for all of your network interfaces. If your present address is 192.168.1.xxx, where 'xxx' does not equal 50, and your subnet mask is 255.255.255.0, then you are ready to configure your IANF2(E). Figure A.1 shows the output of an ipconfig command that shows the "Local Area Connection 2" interface on the 192.168.1.xxx subnet.

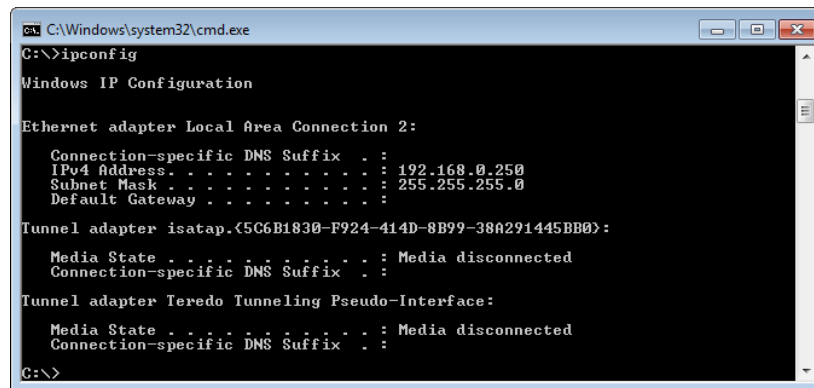


Figure A.1 ipconfig Command

If your present address in not in the 192.168.1.xxx range, type in 'ncpa.cpl' at the command prompt and hit [Enter] on the keyboard.

- ▶ For Vista and Windows 7, this open the *Network Connections* window. Double click on the appropriate interface. In the window that opens, select "Internet Protocol Version 4 (TCP/IP v4)" from the list and then click on the [Properties] button.
- ▶ For Windows 8 and 10, this open the *Network Connections* window. Double click on the appropriate interface. In the window that opens, select "Internet Protocol Version 4 (TCP/IP v4)" from the list and then click on the [Properties] button.

Set the address and subnet mask to appropriate values. (192.168.1.1 and 255.255.255.0 will work for an IANF2(E) that has factory default settings.) The default gateway and DNS server settings can be ignored.

A.4 Test Your Network Interface

Going back to the terminal you opened in the last step, type in 'ping aaa.bbb.ccc.ddd' where 'aaa.bbb.ccc.ddd' is the IP address of the IANF2E that is the network connection for the AnyNET-I/O Stack. The computer will ping the unit and the message "Reply from aaa.bbb.ccc.ddd: bytes=32 time<10ms TTL=255" should appear four times.

If the message "Request timed out." or "Destination host unreachable" appears, then one of four things has occurred:

- You set a new IP address, but have not yet cycled power to the IANF2E that is the network connection for the AnyNET-I/O Stack.
- You did not enter the correct address in the ping command.
- The IP address of the IANF2E that is the network connection for the AnyNET-I/O Stack, is not set correctly.
- The IANF2E that is the network connection for the AnyNET-I/O Stack and the computer are not on the same subnet.

Notes

